G. M. Amdahl
G. A. Blaauw
F. P. Brooks, Jr.

Architecture of the IBM System/360

Abstract: The architecture* of the newly announced IBM System/360 features four innovations:

1. An approach to storage which permits and exploits very large capacities, hierarchies of speeds, read-only storage for microprogram control, flexible storage protection, and simple program relocation.

2. An input/output system offering new degrees of concurrent operation, compatible channel operation, data rates approaching 5,000,000 characters/second, integrated design of hardware and software, a new low-cost, multiple-channel package sharing main-frame hardware, new provisions for device status information, and a standard channel interface between central processing unit and input/output devices.

3. A truly general-purpose machine organization offering new supervisory facilities, powerful logical processing operations, and a wide variety of data formats.

4. Strict upward and downward machine-language compatibility over a line of six models having a performance range factor of 50.

This paper discusses in detail the objectives of the design and the rationale for the main features of the architecture. Emphasis is given to the problems raised by the need for compatibility among central processing units of various size and by the conflicting demands of commercial, scientific, real-time, and logical information processing. A tabular summary of the architecture is shown in the Appendices.

Introduction

The design philosophies of the new general-purpose machine organization for the IBM System/360 are discussed in this paper.† In addition to showing the architecture* of the new family of data processing systems, we point out the various engineering problems encountered in attempts to make the system design compatible, at the program bit level, for large and small models. The compatibility was to extend not only to models of any size but also to their various applications—scientific, commercial, real-time, and so on.

The section that follows describes the objectives of the new system design, i.e., that it serve as a base for new technologies and applications, that it be general-purpose, efficient, and strictly program compatible in all models. The remainder of the paper is devoted to the design problems faced, the alternatives considered, and the decisions made for data format, data and instruction codes, storage assignments, and input/output controls.

Design objectives

The new architecture builds upon but differs from the designs that have gradually evolved since 1950. The evolution of the computer had included, besides major technological improvements, several important systems concepts and developments:

---

* The term *architecture* is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.

† Additional details concerning the architecture, engineering design, programming, and application of the IBM System/360 will appear in a series of articles in the *IBM Systems Journal*.

REPRINTED FROM IBM JOURNAL OF RESEARCH AND DEVELOPMENT, VOL. 8, NO. 2, 1964; ©1964, 2000

0018-8646 / 00 / $5.00 © 2000 IBM

1. Adaptation to business data processing.

2. Growing importance of the total system, especially the input/output aspects.

3. Universal use of assembly programs, compilers, and other metaprograms.

4. Development of magnetic recording on tapes, drums, and disks.

5. Hundred-fold expansion of storage capacities.

6. Adaptation for real-time systems.

During this period most new computer models, from the point of view of their logical structure, were improved, enlarged, or technologically recast versions of the machines developed in the early 1950's. IBM products are not atypical; the evolution has gone from IBM 701 to 7094, 650 to 7074, from 702 to 7080, and from 1401 to 7010.

The system characteristics to be described here, however, are a new approach to logical structure and function, designed for the needs of the next decade as a coordinated set of data processing systems.

- *Advanced concepts*

It was recognized from the start that the design had to embody recent conceptual advances, and hence, if necessary, be incompatible with existing products. To this end, the following premises were considered:

1. Since computers develop into families, any proposed design would have to lend itself to growth and to successor machines.

2. Input/output (I/O) devices make systems specifically useful for given applications. A general method was needed for using I/O devices differing in data rate, access, and function.

3. The real value of an information system is properly measured by answers-per-month, not bits-per-microsecond. The former criterion required specific advances to increase throughput for a given internal speed, to shorten turnaround time for a given throughput, and to make the whole complex of machines and programming systems easier to use.

4. The functions of the central processing unit (CPU) proper are specific to its application only a minor fraction of the time. The functions required by the system for its own operation, e.g., compiling, input/output management, and the addressing of and within complex data structures, use a major share of time. These functions had to be made efficient, and need not be different in machines designed for different applications.

5. The input/output channel and the input/output control program had to be designed for each other.

6. Machine systems had to be capable of supervising themselves, without manual intervention, for both real-time and multiprogrammed, or time-shared, applications. To realize this capability requires: a comprehensive interruption system, tamper-proof storage protection, a protected supervisor program, supervisor-controlled program switching, supervisor control of all input/output (including unit assignment), nonstop operation (no HALT), easy program relocation, simple writing of read-only or unmodified programs, a timer, and interpretive consoles.

7. It must be possible and straightforward to assemble systems with redundant I/O, storages, and CPU's so that the system can operate when modules fail.

8. Storage capacities of more than the commonly available 32,000 words would be required.

9. Certain types of problems require floating-point word length of more than 36 bits.

10. As CPU's become increasingly reliable, built-in thorough checking against hardware malfunction is imperative for all systems, regardless of application.

11. Since the largest servicing problem is diagnosis of malfunction, built-in hardware fault-locating aids are essential to reduce down-times. Furthermore, identification of individual malfunctions and of individual invalidities in program syntax would have to be provided.

- *Open-ended design*

The new design had to provide a dependable base for a decade of customer planning and customer programming, and continuing laboratory developments, whether in technology, application and programming techniques, system configuration, or special requirements.

The various circuit, storage, and input/output technologies used in a system change at different times, causing corresponding changes in their *relative* speeds and costs. To take advantage of these changes, it is desirable that the design permit asynchronous operation of these components with respect to each other.

Changing application and programming techniques would require open-endedness in function. Current trends had to be extrapolated and their consequences anticipated. This anticipation could be achieved by direct provision, e.g., by increasing storage capacities and by using multiple-CPU systems, various new I/O devices, and time sharing. Anticipation might also take the form of generalization of function, as in code-independent scan and translation facilities, or it might consist of judiciously reserving spare bits, operation codes, and blocks of operation codes, for new modes, operations, or sets of operations.

Changing requirements for system configuration would demand not only such approaches as a standard interface between I/O devices and control unit, but also capabilities for a machine to directly sense, control, and respond to other equipment modules via paths outside the normal data routes. These capabilities permit the construction of supersystems that can be dynamically reconfigured under program control, to adapt more precisely to specialized functions or to give graceful degradation.

In many particular applications, some special (and often minor) modification enhances the utility of the system. These modifications (RPQ's), which may correct some shortsightedness of the original design, often embody operations not fully anticipated. In any event, a good general design would obviate certain modifications and accommodate others.

- *General-purpose function*

The machine design would have to provide individual system configurations for large and small, separate and mixed applications as found in commercial, scientific, real-time, data-reduction, communications, language, and logical data processing. The CPU design would have to be facile for each of these applications. Special facilities such as decimal or floating-point arithmetic might be required only for one or another application class and would be offered as options, but they would have to be integral, from the viewpoint of logical structure, with the design.

In particular, the general-purpose objective dictated that:

1. Logical power of great generality would have to be provided, so that all combinations of bits in data entities would be allowed and might be manipulated with operations whose power and utility depend upon the general nature of representations rather than upon any specific selection of them.

2. Operations would have to be code-independent except, of course, where code definition is essential to operation, as in arithmetic. In particular, all bit combinations should be acceptable as data; no combination should exert any control function when it appears in a data stream.

3. The individual bit would have to be separately manipulatable.

4. The general addressing system would have to be able to refer to small units of bits, preferably the unit used for characters.

Further, the implications of general-purpose CPU design for communications-oriented systems indicated a radical departure from current systems philosophy. The conventional CPU, for example, is augmented by an independent stored-program unit (such as the IBM 7750 or 7740) to handle all communications functions. Since the new CPU would easily perform such *logical* functions as code translation and message assembly, communications lines would be attached directly to the I/O channel via a control unit that would perform only character assembly and the electrical line-handling functions.

- *Efficient performance*

The basic measure of a good design is high performance in comparison to other designs having the same cost. This measure cannot be ignored in designing a compatible line. Hence each individual model and systems configuration in the line would have to be competitive with systems that are specialized in function, performance level or both. That this goal is feasible in spite of handicaps introduced by the compatibility requirement was due to the especially important cost savings that would be realized due to compatibility.

- *Intermodel compatibility*

The design had to yield a range of models with internal performance varying from approximately that of the IBM 1401 to well beyond that of the IBM 7030 (STRETCH). As already mentioned, all models would have to be strictly program compatible, upward and downward, at the program bit level.

The phrase "strictly program compatible" requires a more technically precise definition. Here it means that a valid program, whose logic will not depend implicitly upon time of execution and which runs upon configuration **A**, will also run on configuration **B** if the latter includes at least the required storage, at least the required I/O devices, and at least the required optional features. Invalid programs, i.e., those which violate the programming manual, are not constrained to yield the same results on all models. The manual identifies not only the results of all dependable operations, but also those results of exceptional and/or invalid operations that are not dependable. Programs dependent on execution-time will operate compatibly if the dependence is explicit, and, for example, if completion of an I/O operation or the timer are tested.

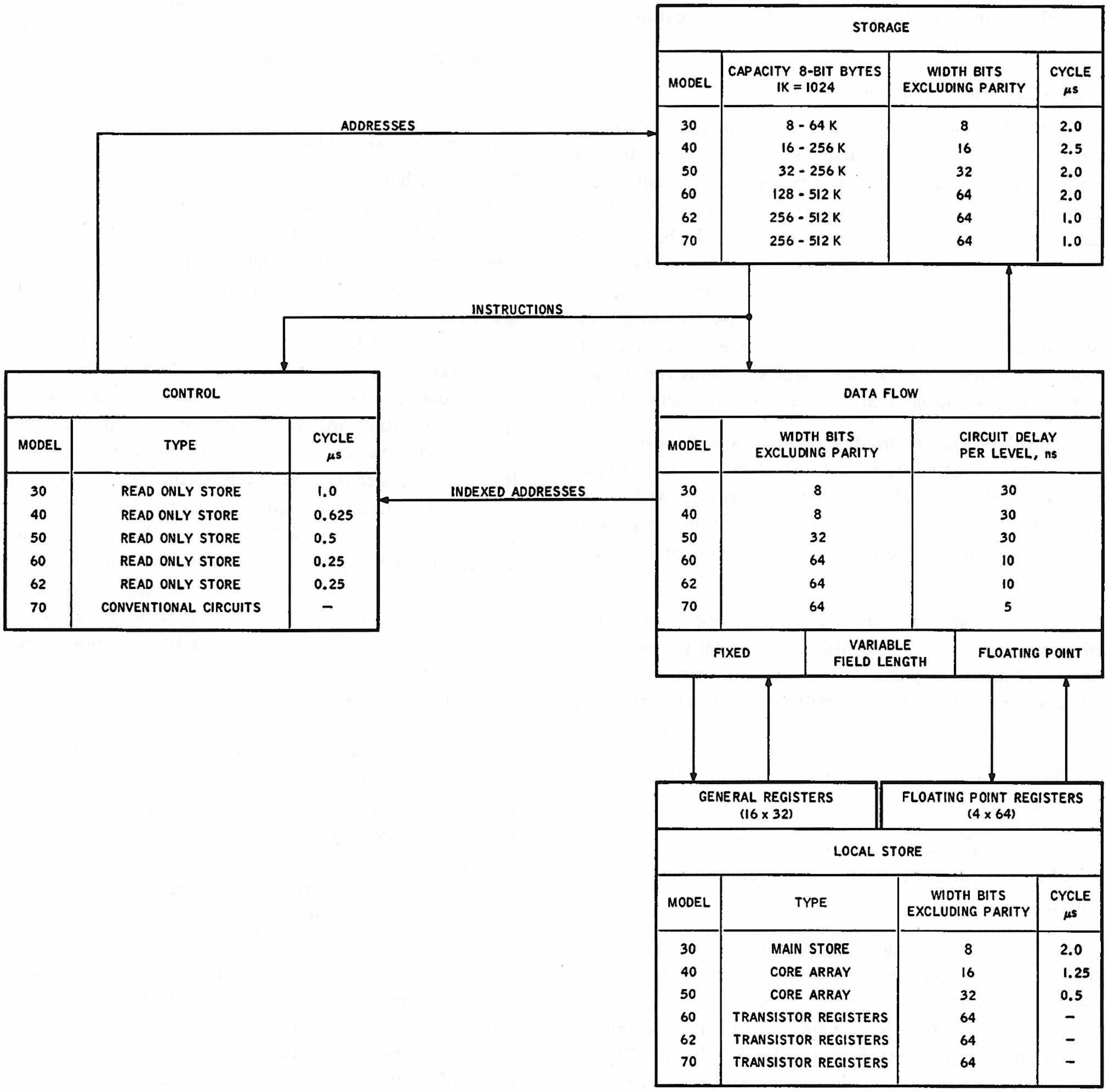
Compatibility would ensure that the user's expanding needs be easily accommodated by any model. Compatibility would also ensure maximum utility of programming support prepared by the manufacturer, maximum sharing of programs generated by the user, ability to use small systems to back up large ones, and exceptional freedom in configuring systems for particular applications.

It required a new concept and mode of thought to make the compatibility objective even conceivable. In the last few years, many computer architects had realized, usually implicitly, that logical structure (as seen by the programmer) and physical structure (as seen by the engineer) are quite different. Thus each may see registers, counters, etc., that to the other are not at all real entities. This was not so in the computers of the 1950's. The *explicit* recognition of the duality of structure opened the way for the compatibility within System/360. The compatibility requirement dictated that the basic architecture had to embrace different technologies, different storage-circuit speed ratios, different data path widths, and different data-flow complexities. The basic machine structure and implementation at the various performance levels are shown in Fig. 1.

The design decisions

Certain decisions for the architectural design became mileposts, because they (a) established prominent characteristics of the System/360, (b) resolved problems concerning the compatibility objective, thus illuminating the essential differences between small models and large, or (c) resolved problems concerning the general-purpose objective, thus illuminating the essential differences among applications. The sections that follow discuss these de-

Figure 1 **Machine structure and implementation.**



STORAGE

MODEL	CAPACITY 8-BIT BYTES 1K = 1024	WIDTH BITS EXCLUDING PARITY	CYCLE µs
30	8 - 64 K	8	2.0
40	16 - 256 K	16	2.5
50	32 - 256 K	32	2.0
60	128 - 512 K	64	2.0
62	256 - 512 K	64	1.0
70	256 - 512 K	64	1.0

CONTROL

MODEL	TYPE	CYCLE µs
30	READ ONLY STORE	1.0
40	READ ONLY STORE	0.625
50	READ ONLY STORE	0.5
60	READ ONLY STORE	0.25
62	READ ONLY STORE	0.25
70	CONVENTIONAL CIRCUITS	—

DATA FLOW

MODEL	WIDTH BITS EXCLUDING PARITY	CIRCUIT DELAY PER LEVEL, ns
30	8	30
40	8	30
50	32	30
60	64	10
62	64	10
70	64	5

FIXED	VARIABLE FIELD LENGTH	FLOATING POINT

GENERAL REGISTERS (16 x 32)	FLOATING POINT REGISTERS (4 x 64)

LOCAL STORE

MODEL	TYPE	WIDTH BITS EXCLUDING PARITY	CYCLE µs
30	MAIN STORE	8	2.0
40	CORE ARRAY	16	1.25
50	CORE ARRAY	32	0.5
60	TRANSISTOR REGISTERS	64	—
62	TRANSISTOR REGISTERS	64	—
70	TRANSISTOR REGISTERS	64	—

cisions, the problems faced, the alternatives considered, and the reasons for the outcome.

• *Data format*

The decision on basic format (which affected character size, word size, instruction field, number of index registers, input-output implementation, instruction set layout, storage capacity, character code, etc.) was whether data length modules should go as 2^n or $3 \cdot 2^n$. Even though many matters of format were considered in the basic choice, we will for convenience treat the major components of the decision as if they were independent.

Character size, 6 vs 4/8. In character size, the fundamental problem is that decimal digits require 4 bits, the alphanumeric characters require 6 bits. Three obvious alternatives were considered — 6 bits for all, with 2 bits wasted on numeric data; 4 bits for digits, 8 for alphanumeric, with 2 bits wasted on alphanumeric; and 4 bits for digits, 6 for alphanumeric, which would require adoption of a 12-bit module as the minimum addressable element. The 7-bit character, which incorporated a binary recoding of decimal digit pairs, was also briefly examined.

The 4/6 approach was rejected because (a) it was desired to have the versatility and power of manipulating character streams and addressing individual characters, even in models where decimal arithmetic is not used, (b) limiting the alphabetic character to 6 bits seemed short-sighted, and (c) the engineering complexities of this approach might well cost more than the wasted bits in the character.

The straight-6 approach, used in the IBM 702–7080 and 1401–7010 families, as well as in other manufacturers' systems, had the advantages of familiar usage, existing I/O equipment, simple specification of field structure, and commensurability with a 48-bit floating-point word and a 24-bit instruction field.

The 4/8 approach, used in the IBM 650–7074 family and elsewhere, had greater coding efficiency, spare bits in the alphabetic set (allowing the set to grow), and commensurability with a 32/64-bit floating-point word and a 16-bit instruction field. Most important of these factors was coding efficiency, which arises from the fact that the use of numeric data in business records is more than twice as frequent as alphanumeric. This efficiency implies, for a given hardware investment, better use of core storage, faster tapes, and more capacious disks.

Floating-point word length, 48 vs 32/64. For large models addition time goes up slowly with word length, and multiplication time rises almost linearly. For small, serial models, addition time rises linearly and multiplication as the square of word length. Input/output time for data files rises linearly. Large machines more often require high precision; small machines more urgently require short operands. For this aspect of the basic format problem, then, definite conflicts arose because of compatibility.

Good data were unavailable on the distribution of required precision by the number of problems or running time. Indeed, accurate measures could not be acquired on such coarse parameters as frequency of double-precision operation on 36-bit and 48-bit machines. The question became whether to force all problems to the longer 48-bit word, or whether to provide 64 to take care of precision-sensitive problems adequately, and either 32 or 36 to give faster speed and better coding efficiency for the rest. The choice was made for the IBM System/360 to have both 64- and 32-bit length floating point. This choice offers the user the option of making the speed/space vs precision trade-off to best suit his requirements. The user of the large models is expected to employ 64-bit words most of the time. The user of the smaller models will find the 32-bit length advantageous in most of his work. All floating-point models have both lengths and operate identically.

Hexadecimal floating-point radix. With no conflicts in questions of large vs small machines, base 16 was selected for floating point. Studies by Sweeney[1] show that the frequency of pre-shift, overflow, and precision-loss post-shift on floating-point addition are substantially reduced by this choice. He has shown that, compared with base 2, the percentage frequency of occurrence of overflow is 5 versus 20, pre-shift is 43 versus 58, and precision-loss post-shift is 11 versus 18. Thus speed is noticeably enhanced. Also, simpler shifting paths, with fewer logic levels, will accomplish a higher proportion of all required pre-shifting in a single pass. For example, circuits shifting 0, 1, 2, 3, or 4 binary places cover 82% of the base 2 pre-shifts. Substantially simpler circuits shifting 0, 1, or 2 hexadecimal places cover 93% of all base 16 pre-shifts. This simplification yields higher speed for the large models and lower cost for the small ones.

The most substantial disadvantage of adopting base 16 is the shift in bit usage from exponent to fraction. Thus, for a given range and a given *minimum* precision, base 16 requires 2 fewer exponent bits and 3 more fraction bits than does base 2. Alternatively and equivalently, rounding and truncation effects are 8 times as large for a given fraction length. For the 64-bit length, this is no problem. For the 32-bit length, with its 24-bit fraction, the minimum precision is reduced to the equivalent of 21 bits. Because the 64-bit length was available for problems where the minimum precision cramped the user, the greater speed and simplicity of base 16 was chosen.

Significance arithmetic. Many schemes yielding an estimate of the significance of computed results have been proposed. One such scheme, a modified form of unnormalized arithmetic, was for a time incorporated in the design. The scheme was finally discarded when simulation runs showed this mode of operation to cost about one hexadecimal digit of actual significance developed, as compared with normalized operation. Furthermore, the significance estimate yielded for a given problem varied substantially with the test data used.

Sign representations. For the fixed-point arithmetic system, which is binary, the two's complement representation for negative numbers was selected. The well-known virtues of this system are the unique representation of zero and the absence of recomplementation. These substantial advantages are augmented by several properties especially useful in address arithmetic, particularly in the large models, where address arithmetic has its own hardware. With two's complement notation, this indexing hardware requires no true/complement gates and thus works faster. In the smaller, serial models, the fact that high-order bits of address arithmetic can be elided without changing the low-order bits also permits a gain in speed. The same truncation property simplifies double-precision calculations. Furthermore, for table calculation, rounding or truncation to an integer changes all variables in the same direction, thus giving a more acceptable distribution than does an absolute-value-plus-sign representation.

The established commercial rounding convention made the use of complement notation awkward for decimal data; therefore, absolute-value-plus-sign is used here. In floating point, the engineering virtues of normalizing only high-order zeros, and of having all zeros represent the smallest possible number, decided the choice in favor of absolute-value-plus-sign.

Variable- versus fixed-length decimal fields. Since the fields of business records vary substantially in length, coding efficiency (and hence tape speed, file capacity, CPU speed, etc.) can be gained by operating directly on variable-length fields. This is easy for serial-by-byte machines, and the IBM 1401–7010 and 702–7080 families are among those so designed. A less flexible structure is more appropriate for a more parallel machine, and the IBM 650–7074 family is among those designed with fixed-word-length decimal arithmetic.

As one would expect, the storage efficiency advantage of the variable data format is diminished by the extra instruction information required for length specification. While the fixed format is preferable for the larger machines, the variable format was adopted because (a) the small commercial users are numerous and only recently trained in variable-format concepts, and (b) the large commercial system is usually I/O limited; hence the internal performance disadvantage of the variable format is more than compensated by the gain in effective tape rate.

Decimal accumulators versus storage-storage operation. A closely related question involving large/small models concerned the use of an accumulator as one of the operands on decimal arithmetic, versus the use of storage locations for all operands and results. This issue is pertinent even after a decision has been made for variable-length fields in storage; for example, it distinguishes IBM 702–7080 arithmetic from that of the IBM 1401–7010 family.

The large models readily afford registers or local stores and get a speed enhancement from using these as accumulators. For the small model, using core storage for logical registers, addition to an accumulator is no faster than addition to a programmer-specified location. Addition of two arbitrary operands and storage of the result becomes LOAD, ADD, STORE, however, and this operation is substantially slower for the small models than the MOVE, ADD sequence appropriate to storage-storage operation. Business arithmetic operations (as hand coded and especially as compiled from COBOL) often take this latter form and rarely occur in strings where intermediate results are profitably held in accumulators. In address arithmetic and floating-point arithmetic, quite the opposite is true.

Field specification: word-marks versus length. Variable-length fields can be specified in the data via delimiter characters or word-marks, or in the instruction via specification of field length or start-finish limits. For business *data*, the word-mark has some slight advantage in storage efficiency: one extra bit per 8-bit character would cost less than 4 extra length bits per 16-bit address. Furthermore, instructions, and hence addresses, usually occupy most core storage space in business computers. However, the word-mark approach implies the use of word-marks on instructions, too, and here the cost is without compensating function. The same is true of all fixed-field data, an important consideration in a general-purpose design. On balance, storage efficiency is about equal; the field specification was put in the instruction to allow all data combinations to be valid and to give easier and more direct programming, particularly since it provides convenient addressing of parts of fields. Length was chosen over limit specification to simplify program relocation and instruction modification.

ASCII vs BCD codes. The selection of the 8-bit character size in 1961 proved wise by 1963, when the American Standards Association adopted a 7-bit standard character code for information interchange (ASCII). This 7-bit code is now under final consideration by the International Standards Organization for adoption as an international standards recommendation. The question became "Why not adopt ASCII as the *only* internal code for System/360?"

The reasons against such exclusive adoption was the widespread use of the BCD code derived from and easily translated to the IBM card code. To facilitate use of both codes, the central processing units are designed with a high degree of code independence, with generalized code translation facilities, and with program-selectable BCD or ASCII modes for code-dependent instructions. Neverthe-

Figure 2a **Extended binary-coded-decimal (BCD) interchange code.**

Bit Positions 01 →	00				01				10				11			
23 → / 4567	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL				BLANK	&	-						>	<	‡	0
0001							/		a	j			A	J		1
0010									b	k	s		B	K	S	2
0011									c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	EOB	UC					f	o	w		F	O	W	6
0111	DEL	IDL	PRE	EOT					g	p	x		G	P	X	7
1000									h	q	y		H	Q	Y	8
1001					.		'	"	i	r	z		I	R	Z	9
1010					?	!		:								
1011					.	$	,	#								
1100					←	*	%	@								
1101					(	)	⌒	'								
1110					+	;	_	=								
1111					⧻	¢	±	✓								

Figure 2b **8-bit representation of the 7-bit American Standard Code for Information Interchange (ASCII).**

Bit Positions 76 →	00				01				10				11			
X5 → / 4321	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL	DC0			BLANK	0					@	P				p
0001	SOM	DC1			!	1					A	Q			a	q
0010	EOA	DC2			"	2					B	R			b	r
0011	EOM	DC3			#	3					C	S			c	s
0100	EQT	DC4 STOP			$	4					D	T			d	t
0101	WRU	ERR			%	5					E	U			e	u
0110	RU	SYNC			&	6					F	V			f	v
0111	BELL	LEM			'	7					G	W			g	w
1000	BKSP	S0			(	8					H	X			h	x
1001	HT	S1			)	9					I	Y			i	y
1010	LF	S2			*	:					J	Z			j	z
1011	VT	S3			+	;					K	[			k	
1100	FF	S4			,	<					L	\			l	
1101	CR	S5			-	=					M	]			m	
1110	SO	S6			.	>					N	↑			n	ESC
1111	SI	S7			/	?					O	←			o	DEL

less, a choice had to be made for the code-sensitive I/O devices and for the programming support, and the solution was to offer both codes, fully supported, as a user option. Systems with either option will, of course, easily read or write I/O media with the other code. The extended BCD interchange code and an 8-bit representation of the 7-bit ASCII are shown in Fig. 2.

Boundary alignment. A major compatibility problem concerned alignment of field boundaries. Different models were to have different widths of storage and data flow, and therefore each model had a different set of preferences. For the 8-bit wide model the characters might have been aligned on character boundaries, with no further constraints. In the 64-bit wide model it might have been preferred to have no fields split between different 64-bit double-words. The general rule adopted (Fig. 3) was that each fixed field must begin at a multiple of its field length, and variable-length decimal and character fields are unconstrained and are processed serially in all models. All models must insure that programmers will adhere to these rules. This policing is essential to prevent the use of technically invalid programs that might work beautifully on small models but not on large ones. Such an outcome would undermine compatibility. The general rule, which has very few and very minor exceptions, is that invalidities defined in the manual are detected in the hardware and cause an interruption. This type of interruption is distinct from an interruption caused by machine malfunctions.

- *Instruction decisions*

Pushdown stack vs addressed registers. Serious consideration was given to a design based on a pushdown accumulator or stack.[2] This plan was abandoned in favor of several registers, each explicitly addressed. Since the advantages of the pushdown organization are discussed in the literature,[3] it suffices here to enumerate the disadvantages which prompted the decision to use an addressed-register organization:

1. The performance advantage of a pushdown stack organization is derived principally from the presence of several fast registers, not from the way they are used or specified.

2. The fraction of "surfacings" of data in the stack which are "profitable," i.e., what was needed next, is about one-half in general use, because of the occurrence of repeated operands (both constants and common factors). This suggests the use of operations such as TOP and SWAP, which respectively copy submerged data to the active positions and assist in clearing submerged data when the information is no longer needed.

3. With TOP's and SWAP's counted, the substantial instruction density gained by the widespread use of implicit addresses is about equalled by that of the same instructions with explicit, but truncated, addresses which specify only the fast registers.

4. In any practical implementation, the depth of the stack has a limit. The register housekeeping eliminated by the pushdown organization reappears as management of a finite-depth stack and as specification of locations of submerged data for TOP's and SWAP's. Further, when part of a full stack must be dumped to make room for new data, it is the *bottom* part, not the active part, which should be dumped.

5. Subroutine transparency, i.e., the ability to use a subroutine recursively, is one of the apparent advantages of the stack. However, the *disadvantage* is that the transparency does not materialize unless additional independent stacks are introduced for addressing purposes.

6. Fitting variable-length fields into a fixed-width stack is awkward.

In the final analysis, the stack organization would have been about break-even for a system intended principally for scientific computing. Here the general-purpose objective weighed heavily in favor of the more flexible addressed-register organization.

Full vs truncated addresses. From the beginning, the major challenge of compatibility lay in storage addressing. It was clear that large models would require storage capacities in the millions of characters. Small (serial) models would require short addresses to conserve precious core space and instruction fetch time. Some help was given by the decision to use register addressing, which reduces address appearances in the instruction stream by a factor approaching 2.

An early decision had dictated that all addresses had to be indexable, and that a mechanism had to be provided for making all programs easily relocatable. The indexing technique had fully proved its worth in current systems.[4] This technique suggested that abundant address size could be attained through a full-sized index register, used as a base. This approach, coupled with a truncated address in the instruction, gives consequent gains in instruction density. The *base-register* approach was adopted, and then augmented, for some instructions, with a second level of indexing.

Now the question was: How much capacity was to be made directly addressable, and how much addressable only via base registers? Some early uses of base register techniques had been fairly unsuccessful, principally because of awkward transitions between direct and base addressing. It was decided to commit the system completely to a base-register technique; the direct part of the address, the *displacement*, was made so small (12 bits, or 4096 characters) that direct addressing is a practical programming technique only on very small models. This

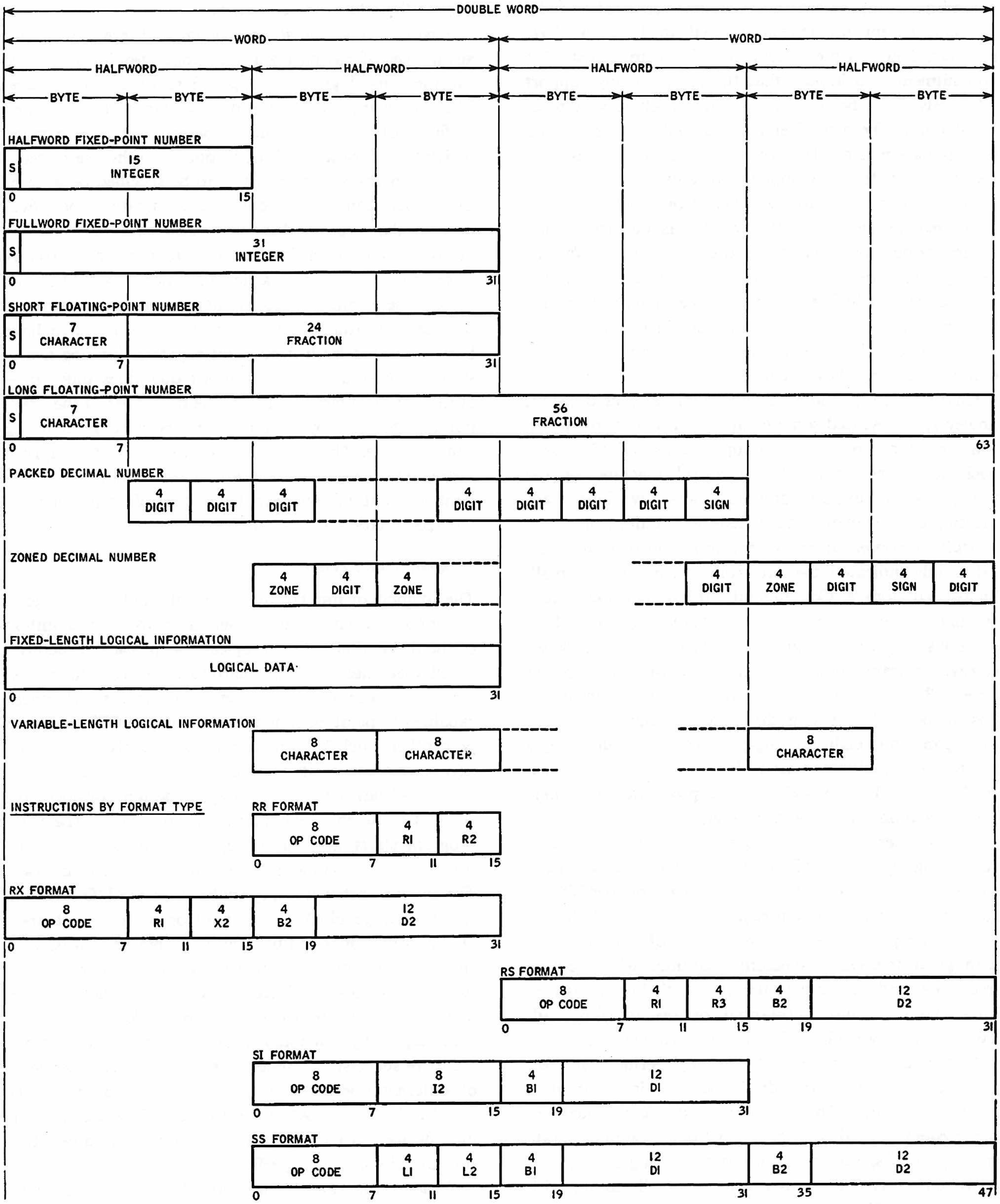


Figure 3 **Boundary alignment of formats.**

VOL. 8, NO. 2, 1964, REPRINT

commitment implies that all programs are location-independent, except for constants used to load the base registers. Thus, all programs can easily be relocated. This commitment also implies that the programming support effectively and efficiently handles the mechanics of base-register use. The assembler automatically constructs and assigns base-plus-displacement addresses as it constructs the symbol table. The compilers not only do this, but also allocate base registers to give efficient programs.

Decimal vs binary addressing. It was decided to use binary rather than decimal addressing, because (a) assembly programs remove the user one level from the address, thus reducing the importance of familiar usage, (b) binary addressing is more efficient in the ratio 3.32/4.00, and (c) table exploitation is easier and more general because any datum can be made into or added to a binary address, yielding a valid address. This decision, however, represented some conflict with past approaches. Machines for purely business applications had often used decimal addressing (in the ancestral machine of the family). Most business computers now have binary addressing or have evolved to mixed-radix addressing.

Multiple accumulators. An extrapolation of technological trends indicated the probable availability of small, high-speed storage. Consequently, the design uses a substantial number of logically identifiable registers, which are physically realized in core storage, local high-speed storage, or transistors, according to the model. There are sixteen 32-bit general-purpose registers and four 64-bit floating-point registers in the logical design, with room for expansion to eight floating-point registers. Surprisingly enough, the multiple-register decision was not a large-small conflict. Each model has an appropriate (and different) mechanization of the same logical design.

Storage hierarchies. Technology promises to yield a continuing spectrum of storage systems whose speed varies inversely with capacity for equal cost-per-bit. Of equal significance, problem requirements naturally follow a matching pattern — small quantities of data are used with great frequency, medium quantities with medium frequency, and very large quantities with low frequency. These facts promise substantial performance/cost advantages if storage hierarchies can be effectively used.

It was decided to accept the engineering, architectural, and programming disciplines required for storage-hierarchy use. The engineer must accommodate in one system several storage technologies, with separate speeds, circuits, power requirements, busing needs, etc., all requiring asynchronous operation of all storage with respect to the CPU. The system programmer must contend with awkward boundaries within total storage capacity and must allocate usage. He must devise addressing for very large capacities, block transfers, and means of handling, indexing across and providing protection across gaps in the addressing sequence.

Separate vs universal accumulators. There are several advantages of having fixed- and floating-point arithmetic use the same logical (as opposed to physical) registers. There are some less obvious disadvantages which weighed in favor of separate accumulator sets. First, in a given register specification (4 bits, in our case) the use of separate sets permits more registers to be specified because of the information implications of the operation code. Second, in the large models instruction execution and the preparation of later instructions are done concurrently in separate units. To use a single register set would couple these closely, and reduce the asynchronous concurrency that can be attained. Historically, index registers have been separated from fixed-point registers, limiting analysis of register allocation to index quantities only. Integration of these facilities brings the full power of the fixed-point arithmetic operation set to bear upon indexing computations. The advantages of the integration appear throughout program execution (even compiler and assembly execution), whereas the register allocation burdens only compilation and assembly.

- *Input/output system*

The method of input/output control would have been a major compatibility problem were it not for the recognition of the distinction between logical and physical structures. Small machines use CPU hardware for I/O functions; large machines demand several independent channels, capable of operating concurrently with the CPU and with each other. Such large-machine channels often each contain more components than an entire small system.

Channel instructions. The logical design considers the channel as an independently operating entity. The CPU program starts the channel operation by specifying the beginning of a channel program and the unit to be used. The channel instructions, specialized for the I/O function, specify storage blocks to be read or written, unit operations, conditional and unconditional branches within the channel program, etc. When the channel program ends, the CPU program is interrupted, and complete channel and device status information are available.

An especially valuable feature is *command chaining*, the ability of successive channel instructions to give a sequence of different operations to the unit, such as SEARCH, READ, WRITE, READ FOR CHECK. This feature permits devices to be reinstructed in very short times, thus substantially enhancing effective speed.

Standard interface. The generalization of the communication between the central processing unit and an input/output device has yielded a channel which presents a standard interface to the device control unit. This interface was achieved by making the channel design transparent, passing not only data, but also control and status information between storage and device. All functions peculiar to the device are placed in the control unit. The interface requires a total of 29 lines and is made independent of time through the use of interlocking signals.

Implementation. In small models, the flow of data and control information is time-shared between the CPU and the channel function. When a byte of data appears from an I/O device, the CPU is seized, dumped, used and restored. Although the maximum data rate handled is lower (and the interference with CPU computation higher) than with separate hardware, the function is identical.

Once the channel becomes a conceptual entity, using time-shared hardware, one may have a large number of channels at virtually no cost save the core storage space for the governing control words. This kind of *multiplex channel* embodies up to 256 conceptual channels, all of which may be concurrently operating, when the total data rate is within acceptable limits. The multiplexing constitutes a major advance for communications-based systems.

Conclusion

This paper has shown how the design features were chosen for the logical structure of the six models that comprise the IBM System/360. The rationale has been given for the adoption of the data formats, the instruction set, and the input/output controls. The main features of the new machine organization are its general-purpose utility for many types of data processing, the new approaches to large-capacity storage, and the machine-language compatibility among the six models.

The contributions discussed in this paper may be summarized as follows:

1. The relative independence of logical structure and physical realization permits efficient implementation at various levels of performance.

2. Tasks that are common to operating a system for most applications require a complement of instructions and system functions that may serve as a base for the addition of application-oriented functions.

3. The formats, instructions, register assignment, and over-all functions such as protection and interruption of a computer can be so defined that they apply to many levels of performance and that they permit diverse specialization for particular applications.

It is hoped that the discussions of these design features will shed some light on the present and future needs of data processing system organization.

Appendices

The design resulting from the decision process sketched above is tabulated in five appendices showing formats, data and instruction codes, storage assignments and interruption action. (*Appendices 1 through 5 appear on the following four pages.*)

Acknowledgments

The implementation of System/360 depends upon diverse developments by many colleagues. The most important of these developments were glass-encapsulated semi-integrated semiconductor components, printed circuit back-panels and interconnections, new memories, read-only storages and microprogram techniques, new I/O devices, and a new level and approach to software support.

The scope of the compatibility objective and of the whole System/360 undertaking was largely due to B. O. Evans, Data Systems Division Vice-President—Development.

References

1. D. W. Sweeney, "An Analysis of Floating-Point Addition and Shifting," to be published in the *IBM Systems Journal.*
2. See, for example, R. S. Barton, "A New Approach to the Functional Design of a Digital Computer," *Proc. WJCC* **19**, 393–396 (1961).
3. F. P. Brooks, Jr., "Recent Developments in Computer Organization," *Advances in Electronics* **18**, 45–64 (1963).
4. G. A. Blaauw, "Indexing," in *Planning a Computer System*, W. Buchholz, ed., McGraw-Hill Book Company Inc., 1962, pp. 150–178.

Received January 21, 1964

VOL. 8, NO. 2, 1964, REPRINT

Appendix 1 **All operation codes are shown in the following table. The 8-bit codes are grouped by the main classes, such as fixed-point arithmetic, floating-point arithmetic and logical operations. The codes are furthermore grouped according to the five main instruction formats RR (register-register), RX (register-indexed storage location), RS (register-storage), SI (storage-immediate information) and SS (storage-storage).**

OPERATION CODES

FORMAT	RR	RR	RR	RR
CLASS	BRANCHING AND STATUS SWITCHING	FIXED-POINT FULLWORD, AND LOGICAL	FLOATING-POINT LONG	FLOATING-POINT SHORT
xxxx	0000xxxx	0001xxxx	0010xxxx	0011xxxx
0000		LOAD POSITIVE	LOAD POSITIVE	LOAD POSITIVE
0001		LOAD NEGATIVE	LOAD NEGATIVE	LOAD NEGATIVE
0010		LOAD AND TEST	LOAD AND TEST	LOAD AND TEST
0011		LOAD COMPLEMENT	LOAD COMPLEMENT	LOAD COMPLEMENT
0100	SET PROGRAM MASK	AND	HALVE	HALVE
0101	BRANCH AND LINK	COMPARE LOGICAL		
0110	BRANCH ON COUNT	OR		
0111	BRANCH/CONDITION	EXCLUSIVE OR		
1000	SET KEY	LOAD	LOAD	LOAD
1001	INSERT KEY	COMPARE	COMPARE	COMPARE
1010	SUPERVISOR CALL	ADD	ADD N	ADD N
1011		SUBTRACT	SUBTRACT N	SUBTRACT N
1100		MULTIPLY	MULTIPLY	MULTIPLY
1101		DIVIDE	DIVIDE	DIVIDE
1110		ADD LOGICAL	ADD U	ADD U
1111		SUBTRACT LOGICAL	SUBTRACT U	SUBTRACT U

FORMAT	RX	RX	RX	RX
CLASS	FIXED-POINT HALFWORD AND BRANCHING	FIXED-POINT FULLWORD AND LOGICAL	FLOATING-POINT LONG	FLOATING-POINT SHORT
xxxx	0100xxxx	0101xxxx	0110xxxx	0111xxxx
0000	STORE	STORE	STORE	STORE
0001	LOAD ADDRESS			
0010	STORE CHARACTER			
0011	INSERT CHARACTER			
0100	EXECUTE	AND		
0101	BRANCH AND LINK	COMPARE LOGICAL		
0110	BRANCH ON COUNT	OR		
0111	BRANCH/CONDITION	EXCLUSIVE OR		
1000	LOAD	LOAD	LOAD	LOAD
1001	COMPARE	COMPARE	COMPARE	COMPARE
1010	ADD	ADD	ADD N	ADD N
1011	SUBTRACT	SUBTRACT	SUBTRACT N	SUBTRACT N
1100	MULTIPLY	MULTIPLY	MULTIPLY	MULTIPLY
1101		DIVIDE	DIVIDE	DIVIDE
1110	CONVERT-DECIMAL	ADD LOGICAL	ADD U	ADD U
1111	CONVERT-BINARY	SUBTRACT LOGICAL	SUBTRACT U	SUBTRACT U

Appendix 2 continued

from addressable registers and storage. The PSW is stored upon interruption. The Channel Command Word controls input/output operation and sequencing. The commands which may be given to the channel are listed as part of the table. The Channel Address Word is used to initiate input/output sequencing. The Channel Status Word indicates the channel status at the completion of an input/output operation or, when specified, during an I/O operation.

CONTROL WORD FORMATS

```
Base and Index Registers

┌0-0-0-0-0-0-0-0-0-0-1-1-1-1-1-1-1-1-1-1-2-2-2-2-2-2-2-2-2-2-3-3┐
|               |                                               |
|       *       |              BASE ADDRESS OR INDEX            |
|               |                                               |
└0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1┘

 0 -  7 Ignored
 8 - 31 Base address or index


Program Status Word

┌0-0-0-0-0-0-0-0-0-0-1-1-1-1-1-1-1-1-1-1-2-2-2-2-2-2-2-2-2-2-3-3┐---
|               |       |       |                               |
| SYSTEM MASK   |  KEY  |A M W P|       INTERRUPTION CODE       |
|               |       |       |                               |
└0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1┘---

---3-3-3-3-3-3-3-3-4-4-4-4-4-4-4-4-4-4-5-5-5-5-5-5-5-5-5-5-6-6-6-6┐
   |   |   |       |                                               |
   |ILC| CC|PROGRAM|              INSTRUCTION ADDRESS              |
   |   |   | MASK  |                                               |
---2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3┘

 0 -  7 System mask
    0    Multiplexor channel mask
    1    Selector channel 1 mask
    2    Selector channel 2 mask
    3    Selector channel 3 mask
    4    Selector channel 4 mask
    5    Selector channel 5 mask
    6    Selector channel 6 mask
    7    External mask
 8 - 11 Protection key
12      ASCII mode (A)
13      Machine check mask (M)
14      Wait state (W)
15      Problem state (P)
16 - 31 Interruption code
32 - 33 Instruction length code (ILC)
34 - 35 Condition code (CC)
36 - 39 Program mask
   36    Fixed-point overflow mask
   37    Decimal overflow mask
   38    Exponent underflow mask
   39    Significance mask
40 - 63 Instruction address
```

Appendix 1 continued

FORMAT CLASS	RS,SI BRANCHING, STATUS SWITCHING AND SHIFTING	RS,SI FIXED-POINT, LOGICAL, AND INPUT/OUTPUT		
xxxx	1000xxxx	1001xxxx	1010xxxx	1011xxxx
0000	SET SYSTEM MASK	STORE MULTIPLE		
0001		TEST UNDER MASK		
0010	LOAD PSW	MOVE		
0011	DIAGNOSE			
0100	WRITE DIRECT	AND		
0101	READ DIRECT	COMPARE LOGICAL		
0110	BRANCH/HIGH	OR		
0111	BRANCH/LOW-EQUAL	EXCLUSIVE OR		
1000	SHIFT RIGHT SL	LOAD MULTIPLE		
1001	SHIFT LEFT SL			
1010	SHIFT RIGHT S			
1011	SHIFT LEFT S			
1100	SHIFT RIGHT DL	START I/O		
1101	SHIFT LEFT DL	TEST I/O		
1110	SHIFT RIGHT D	HALT I/O		
1111	SHIFT LEFT D	TEST CHANNEL		

FORMAT CLASS		SS LOGICAL		SS DECIMAL
xxxx	1100xxxx	1101xxxx	1110xxxx	1111xxxx
0000				
0001		MOVE NUMERIC		MOVE W OFFSET
0010		MOVE		PACK
0011		MOVE ZONE		UNPACK
0100		AND		
0101		COMPARE LOGICAL		
0110		OR		
0111		EXCLUSIVE OR		
1000				ZERO AND ADD
1001				COMPARE
1010				ADD
1011				SUBTRACT
1100		TRANSLATE		MULTIPLY
1101		TRANSLATE AND TEST		DIVIDE
1110		EDIT		
1111		EDIT AND MARK		

Legend
N = Normalized
SL = Single logical
DL = Double logical
U = Unnormalized
S = Single
D = Double

Appendix 2 **The formats of all control words required for CPU and channel operation are shown in the following table. The base and index registers provide 24 bits of address and are specified by the B and X fields of instructions. The Program Status Word controls instruction sequencing and indicates the complete CPU status apart**

Appendix 2 continued

```
    Channel Command Word

┌0-0-0-0-0-0-0-0-0-0-1-1-1-1-1-1-1-1-1-1-2-2-2-2-2-2-2-2-2-2-3-3┐----
|               |                                               |
| COMMAND CODE  |                  DATA ADDRESS                 |
|               |                                               |
└0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1┘----

    ----┌3-3-3-3-3-3-3-3-4-4-4-4-4-4-4-4-4-4-5-5-5-5-5-5-5-5-5-5-6-6-6-6┐
        I         I     I               I                               I
        I  FLAGS  I0 0 0I       *       I             COUNT             I
        I         I     I               I                               I
    ----└2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3┘

 0 -  7 Command code
 8 - 31 Data address
32 - 36 Command flags
   32     Chain data flag
   33     Chain command flag
   34     Suppress length indication flag
   35     Skip flag
   36     Program-controlled interruption flag
37 - 39 Zero
40 - 47 Ignored
48 - 63 Count
```

```
    Channel Address Word

┌0-0-0-0-0-0-0-0-0-0-1-1-1-1-1-1-1-1-1-1-2-2-2-2-2-2-2-2-2-2-3-3┐
|       |       |                                               |
|  KEY  |0 0 0 0|                COMMAND ADDRESS                |
|       |       |                                               |
└0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1┘

 0 -  3 Protection key
 4 -  7 Zero
 8 - 31 Command address
```

```
Channel Status Word

┌0-0-0-0-0-0-0-0-0-0-1-1-1-1-1-1-1-1-1-1-2-2-2-2-2-2-2-2-2-2-3-3┐----
|       |       |                                               |
|  KEY  |0 0 0 0|                COMMAND ADDRESS                |
|       |       |                                               |
└0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1┘----

    ----┌3-3-3-3-3-3-3-3-4-4-4-4-4-4-4-4-4-4-5-5-5-5-5-5-5-5-5-5-6-6-6-6┐
        |                               |                               I
        |            STATUS             |             COUNT             I
        |                               |                               I
    ----└2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3-4-5-6-7-8-9-0-1-2-3┘

 0 -  3 Protection key
 4 -  7 Zero
 8 - 31 Command address
32 - 47 Status
   32     Attention
   33     Status modifier
   34     Control unit end
   35     Busy
   36     Channel end
   37     Device end
   38     Unit check
   39     Unit exception
   40     Program-controlled interruption
```

(continued overleaf)

Appendix 2 continued

41	Incorrect length
42	Program check
43	Protection check
44	Channel data check
45	Channel control check
46	Interface control check
47	Chaining check
48 - 63	Count

Appendix 3 **All permanently assigned storage locations are shown in this table. These locations are addressed by the CPU and I/O channels during initial program loading, during interruptions and in order to update the timer. During initial program loading 24 bytes are read from a specified input device into locations 0 to 23. This information is subsequently used as CCW's to specify the locations of further input information and as a PSW to control CPU operation after the loading operation is completed. During an interruption the current PSW is stored in the "old" location and the PSW from the "new" location is obtained as the next PSW. The timer is counted down and provides an interrupt when zero is passed. All permanently assigned locations may also be addressed by the program.**

PERMANENT STORAGE ASSIGNMENT

	ADDRESS	LENGTH	PURPOSE
0	0000 0000	double word	Initial program loading PSW
8	0000 1000	double word	Initial program loading CCW1
16	0001 0000	double word	Initial program loading CCW2
24	0001 1000	double word	External old PSW
32	0010 0000	double word	Supervisor call old PSW
40	0010 1000	double word	Program old PSW
48	0011 0000	double word	Machine old PSW
56	0011 1000	double word	Input/output old PSW
64	0100 0000	double word	Channel status word
72	0100 1000	word	Channel address word
76	0100 1100	word	Unused
80	0101 0000	word	Timer
84	0101 0100	word	Unused
88	0101 1000	double word	External new PSW
96	0110 0000	double word	Supervisor call new PSW
104	0110 1000	double word	Program new PSW
112	0111 0000	double word	Machine new PSW
120	0111 1000	double word	Input/output new PSW
128	1000 0000		Diagnostic scan-out area*

---

* The size of the diagnostic scan-out area depends upon the particular model and I/O channels.

Appendix 4 continued

Legend

available	Unit and channel available
busy	Unit or channel busy
carry	A carry out of the sign position occurs
complete	Last result byte nonzero
CSW ready	Channel status word ready for test or interruption
CSW stored	Channel status word stored
equal	Operands compare equal
F	Fullword
g zero	Result is greater than zero
H	Halfword
halted	Data transmission stopped. Unit in halt-reset mode
high	First operand compares high
incomplete	Nonzero result byte; not last
L	Long precision
l zero	Result is less than zero
low	First operand compares low
mixed	Selected bits are both zero and one
not oper	Unit or channel not operational
not working	Unit or channel not working
not zero	Result is not all zero
one	Selected bits are one
overflow	Result overflows
S	Short precision
stopped	Data transmission stopped
working	Unit or channel working
zero	Result or selected bits are zero

Note

The condition code also may be changed by LOAD PSW, SET SYSTEM MASK, DIAGNOSE, and by an interruption.

Appendix 5 **All interruptions which may occur are shown in the following table. Indicated here are the code in the old PSW which identifies the source of the interruption, the mask bits which may be used to prevent an interruption, and the manner in which instruction execution is affected. The instruction to be performed next if the interruption had not occurred is indicated in the instruction address field of the old PSW. The length of the preceding instructions, if available, is shown in the instruction length code, ILC, as is further detailed in the table.**

INTERRUPTION ACTION

INTERRUPTION SOURCE IDENTIFICATION	INTERRUPTION CODE PSW BITS 16-31	MASK BITS	ILC SET	INSTRUCTION EXECUTION
Input/Output (old PSW 56, new PSW 120, priority 4)				
Multiplexor channel	00000000 aaaaaaaa	0	x	complete
Selector channel 1	00000001 aaaaaaaa	1	x	complete
Selector channel 2	00000010 aaaaaaaa	2	x	complete

Appendix 4 All instructions which set the condition code (bits 32 and 33 of the PSW) are listed in the following table. All other instructions leave the condition code unchanged. The condition code determines the outcome of a BRANCH ON CONDITION instruction. The four-bit mask contained in this instruction specifies which code settings will cause the branch to be taken.

CONDITION CODE SETTING

	0	1	2	3
Fixed-Point Arithmetic				
ADD H/F	zero	l zero	g zero	overflow
ADD LOGICAL	zero	not zero	zero,carry	carry
COMPARE H/F	equal	low	high	--
LOAD AND TEST	zero	l zero	g zero	--
LOAD COMPLEMENT	zero	l zero	g zero	overflow
LOAD NEGATIVE	zero	l zero	--	--
LOAD POSITIVE	zero	--	g zero	overflow
SHIFT LEFT DOUBLE	zero	l zero	g zero	overflow
SHIFT LEFT SINGLE	zero	l zero	g zero	overflow
SHIFT RIGHT DOUBLE	zero	l zero	g zero	--
SHIFT RIGHT SINGLE	zero	l zero	g zero	--
SUBTRACT H/F	zero	l zero	g zero	overflow
SUBTRACT LOGICAL	--	not zero	zero,carry	carry
Decimal Arithmetic				
ADD DECIMAL	zero	l zero	g zero	overflow
COMPARE DECIMAL	equal	low	high	--
SUBTRACT DECIMAL	zero	l zero	g zero	overflow
ZERO AND ADD	zero	l zero	g zero	overflow
Floating-Point Arithmetic				
ADD NORMALIZED S/L	zero	l zero	g zero	overflow
ADD UNNORMALIZED S/L	zero	l zero	g zero	overflow
COMPARE S/L	equal	low	high	--
LOAD AND TEST S/L	zero	l zero	g zero	--
LOAD COMPLEMENT S/L	zero	l zero	g zero	--
LOAD NEGATIVE S/L	zero	l zero	--	--
LOAD POSITIVE S/L	zero	--	g zero	--
SUBTRACT NORMALIZED S/L	zero	l zero	g zero	overflow
SUBTRACT UNNORMALIZED S/L	zero	l zero	g zero	overflow
Logical Operations				
AND	zero	not zero	--	--
COMPARE LOGICAL	equal	low	high	--
EDIT	zero	l zero	g zero	--
EDIT AND MARK	zero	l zero	g zero	--
EXCLUSIVE OR	zero	not zero	--	--
OR	zero	not zero	--	--
TEST UNDER MASK	zero	mixed	--	one
TRANSLATE AND TEST	zero	incomplete	complete	--
Input-Output Operations				
HALT I/O	not working	halted	stopped	not oper
START I/O	available	CSW stored	busy	not oper
TEST CHANNEL	not working	CSW ready	working	not oper
TEST I/O	available	CSW stored	working	not oper

Appendix 5 continued

Selector channel 3	00000011 aaaaaaaa	3	x	complete
Selector channel 4	00000100 aaaaaaaa	4	x	complete
Selector channel 5	00000101 aaaaaaaa	5	x	complete
Selector channel 6	00000110 aaaaaaaa	6	x	complete

Program (old PSW 40, new PSW 104, priority 2)

Operation	00000000 00000001		1,2,3	suppress
Privileged operation	00000000 00000010		1,2	suppress
Execute	00000000 00000011		2	suppress
Protection	00000000 00000100		0,2,3	suppress/terminate
Addressing	00000000 00000101		0,1,2,3	suppress/terminate
Specification	00000000 00000110		1,2,3	suppress
Data	00000000 00000111		2,3	terminate
Fixed-point overflow	00000000 00001000	36	1,2	complete
Fixed-point divide	00000000 00001001		1,2	suppress/complete
Decimal overflow	00000000 00001010	37	3	complete
Decimal divide	00000000 00001011		3	suppress
Exponent overflow	00000000 00001100		1,2	terminate
Exponent underflow	00000000 00001101	38	1,2	complete
Significance	00000000 00001110	39	1,2	complete
Floating-point divide	00000000 00001111		1,2	suppress

Supervisor Call (old PSW 32, new PSW 96, priority 2)

Instruction bits	00000000 rrrrrrrr		1	complete

External (old PSW 24, new PSW 88, priority 3)

External signal 1	00000000 xxxxxxx1	7	x	complete
External signal 2	00000000 xxxxxx1x	7	x	complete
External signal 3	00000000 xxxxx1xx	7	x	complete
External signal 4	00000000 xxxx1xxx	7	x	complete
External signal 5	00000000 xxx1xxxx	7	x	complete
External signal 6	00000000 xx1xxxxx	7	x	complete
Interrupt key	00000000 x1xxxxxx	7	x	complete
Timer	00000000 1xxxxxxx	7	x	complete

Machine Check (old PSW 48, new PSW 112, priority 1)

Machine malfunction	00000000 00000000	13	x	terminate

Legend

a	Device address bits
r	Bits of R1 and R2 field of SUPERVISOR CALL
x	Unpredictable

INSTRUCTION LENGTH RECORDING

INSTRUCTION LENGTH CODE	PSW BITS 32-33	INSTRUCTION BITS 0-1	INSTRUCTION LENGTH	INSTRUCTION FORMAT
0	00		Not available	
1	01	00	One halfword	RR
2	10	01	Two halfwords	RX
2	10	10	Two halfwords	RS or SI
3	11	11	Three halfwords	SS

Gene M. Amdahl

B.S. in Physics 1948, South Dakota State College; M.S. in Physics, 1949, and Ph.D., 1952, University of Wisconsin. Between 1952 and 1956 he was associated with the IBM Development Laboratory at Poughkeepsie. During the period he was project engineer and systems designer of the IBM 704 and STRETCH computers. From 1956 to 1960 Dr. Amdahl worked at the Ramo Wooldridge Corporation and at Aeronutronic Systems, Inc., where he was manager of Data Processing Engineering. He rejoined IBM in Research in 1960 as Director of Experimental Machines. He has been Advanced Systems Design Manager in the Data Systems Division, Poughkeepsie, and is currently Technical Manager of Large Scientific Systems.

Gerrit A. Blaauw

B.S. in Electrical Engineering, Lafayette College, 1948; Ph.D. in Applied Science, Harvard University, 1952. While at Harvard, was a member of the staff of the Computation Laboratory and participated in the design of the Mark III and Mark IV calculators. From 1952–1955 was a member of the staff of the Mathematics Center in Amsterdam, Netherlands, where he cooperated in the design of the ARRA and FERTA computers. Joined IBM in 1955 at the Poughkeepsie Product Development Laboratory. He has been engaged in the systems planning of various machine projects in Data Systems Division. Is a member of the ACM, IEEE, and Sigma Xi.

Frederick P. Brooks, Jr.

A.B. in Physics, Duke University, 1953. Received S.M., 1955 and Ph.D., 1956, Harvard University, for graduate work in design and application of computers. From 1956 to 1959, Dr. Brooks participated in the planning of the STRETCH and HARVEST computers at the IBM Product Development Laboratory in Poughkeepsie. From 1959 to 1960 he studied computer organization theory at IBM Yorktown Research Center. At present he is IBM Processor Manager, in Poughkeepsie, and is responsible for the specification and development of new computers. Member of IEEE, ACM, Phi Beta Kappa, and Sigma Xi.

VOL. 8, NO. 2, 1964, REPRINT

Form A22-6821-1
File No. S360-01

IBM Systems Reference Library

IBM System/360 Principles of Operation

This manual is a comprehensive presentation of the characteristics, functions, and features of the IBM System/360. The material is presented in a direct manner, assuming that the reader has a basic knowledge of IBM data processing systems and has read the IBM System/360 Systems Summary, Form A22-6810. The manual is useful for individual study, as an instruction aid, and as a machine reference manual.

The manual defines System/360 operating principles, central processing unit, instructions, system control panel, branching, status switching, interruption system, and input/output operations.

Descriptions of specific input/output devices used with System/360 appear in separate publications. Also, details unique to each model of the System/360 appear in separate publications.

MAJOR REVISION (June 1964)
This publication, Form A22-6821-1, obsoletes
Forms A22-6821-0 and N22-0136.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the contents of this publication to:
IBM Corporation, Customer Manuals, Dept. B98, PO Box 390, Poughkeepsie, N. Y. 12602

ou e word in a pair of
ftu left or right. A pair
CONVERT TO BINARY and
vic s transition between
un er base) without the
r loading and storing in-
e switching.

d for processes requiring
w n the source input
T s type of processing
ercial applications, par-
of problem-oriented lan-
d umber of arithmetic
it n of data, radix con-
ry and back to decimal
r gisters for intermedi-
e er storage-to-storage
arithmetic is provided.
s are located in storage
ldition, subtraction,
o parison.
l as signed integers with
f m one to 16 bytes
r d in true form.
epresented in the four-
by 0000-1001, respec-
t valid as digits and
0 and 1101 represent
are interpreted as plus
decimal arithmetic de-
ferred (Figure 7).
decimal interchange
e codes are 1100 and
a ed to eight bits, is
a d 1011. The choice
determined by a mode

se ed by four-bit bin-
d wo to a byte. They
gth and are accompa-
t four bits of the low-

n

order byte. Operand fields may be located on any byte boundary, and may have length up to 31 digits and sign. Operands participating in an operation have independent lengths. Packing of digits within a byte (Figure 8) and of variable-length fields within storage results in efficient use of storage, in increased arithmetic performance, and in an improved rate of data transmission between storage and files.

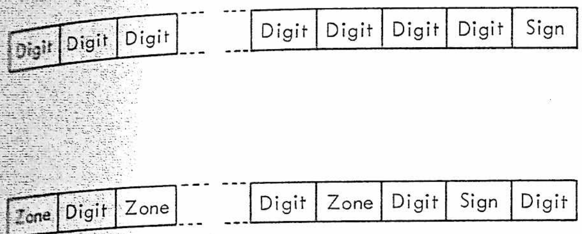


Figure 8. Packed and Zoned Decimal Number Formats

Decimal numbers may also appear in a zoned format as a subset of the eight-bit alphameric character set (Figure 8). This representation is required for character-set sensitive I/O devices. The zoned format is not used in decimal arithmetic operations. Instructions are provided for packing and unpacking decimal numbers so that they may be changed from the zoned to the packed format and vice versa.

Floating-Point Arithmetic

Floating-point numbers occur in either of two fixed-length formats – short or long. These formats differ only in the length of the fractions (Figure 9).

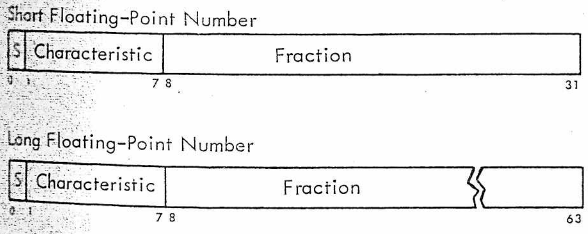


Figure 9. Short and Long Floating-Point Number Formats

Operands are either 32 or 64 bits long. The short length, equivalent to seven decimal places of precision, permits a maximum number of operands to be placed in storage and gives the shortest execution times. The long length, used when higher precision is desired, gives up to 17 decimal places of precision, thus eliminating most requirements for double-precision arithmetic.

The operand lengths, being powers of two, permit maximum efficiency in the use of binary addressing and in matching the physical word sizes of storage. Floating-point arithmetic is designed to allow easy transition between the two formats.

The fraction of a floating-point number is expressed in hexadecimal (base 16) digits, each consisting of four binary bits and having the values 0-15. In the short format, the fraction consists of six hexadecimal digits occupying bits 8-31. In the long format the fraction has 14 hexadecimal digits occupying bits 8-63.

The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit. To provide the proper magnitude for the floating-point number, the fraction is considered to be multiplied by a power of 16. The characteristic portion, bits 1-7 of both formats, is used to indicate this power. The characteristic is treated as an excess 64 number with a range from –64 through +63, and permits representation of decimal numbers with magnitudes in the range of 10^{-78} to 10^{75}.

Bit position 0 in either format is the sign (S) of the fraction. The fraction of negative numbers is carried in true form.

Four 64-bit floating-point registers are provided. Arithmetic operations are performed with one operand in a register and another either in a register or from storage. The result, developed in a register, is generally of the same length as the operands. The availability of several floating-point registers eliminates much storing and loading of intermediate results.

Logical Operations

Logical information is handled as fixed-length and variable-length data. It is subject to such operations as comparison, translation, editing, bit testing, and bit setting.

When used as a fixed-length operand, logical information can consist of either one, four, or eight bytes and is processed in the general registers.

A large portion of logical information consists of alphabetic or numeric character codes, called *alphameric data,* and is used for communication with character-set sensitive I/O devices. This information has the variable-field-length format and can consist of up to 256 bytes (Figure 10). It is processed storage to storage, left to right, an eight-bit byte at a time.

The CPU can handle any eight-bit character set, although certain restrictions are assumed in the decimal arithmetic and editing operations. However, all character-set sensitive I/O equipment will assume either the extended binary-coded-decimal interchange code

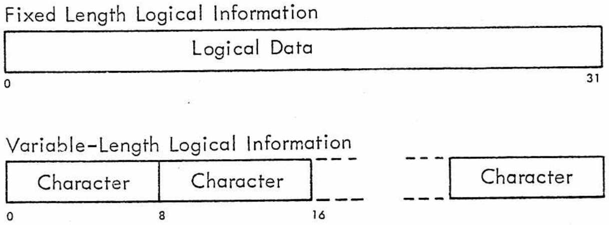

Fixed Length Logical Information
Logical Data
0
31
Variable-Length Logical Information
Character
Character
Character
0
8
16

Figure 10. Fixed-Length and Variable-Length Logical Information

(EBCDIC) (Figure 11) or the American Standard Code for Information Interchange (ASCII) extended to eight bits, referred to as ASCII-8 in this manual (Figure 12). The numbering convention for the bit positions (upper left-hand corner of Figures 12 and 13) within a character differ for each of the codes. The conventions are as follows:

	BIT POSITIONS
EBCDIC	01234567
ASCII-8	76x54321

The preferred codes do not have a graphic defined for all 256 eight-bit codes. When it is desirable to represent all possible bit patterns, a hexadecimal representation may be used instead of the preferred eight-bit code. The hexadecimal representation uses one graphic for a four-bit code, and therefore, two graphics for an eight-bit byte. The graphics 0-9 are us for codes 0000-1001; the graphics A-F are used f codes 1010-1111.

Program Execution

The CPU program consists of instructions, index wo and control words specifying the operations to be p formed. This information resides in main storage a general registers, and may be operated upon as dat

Instruction Format

The length of an instruction format can be one, tw or three halfwords. It is related to the number of sto age addresses necessary for the operation. An instru tion consisting of only one halfword causes no refe ence to main storage. A two-halfword instruction pro vides one storage-address specification; a three-hal word instruction provides two storage-address speci cations. All instructions must be located in storage o integral boundaries for halfwords. Figure 13 show five basic instruction formats.

The five basic instruction formats are denoted b the format codes RR, RX, RS, SI, and SS. The forma codes express, in general terms, the operation to b performed. RR denotes a register-to-register operation RX, a register-to-indexed-storage operation; RS, a regis-

Bit Positions → 01	00				01				10				11			
→ 23 → 4567	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL				b̸ blank	&	-						>	<	‡	0
0001							/		a	j			A	J		1
0010									b	k	s		B	K	S	2
0011									c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	EOB	UC					f	o	w		F	O	W	6
0111	DEL	IDL	PRE	EOT					g	p	x		G	P	X	7
1000									h	q	y		H	Q	Y	8
1001					.		,	"	i	r	z		I	R	Z	9
1010					?	!		:								
1011					.	$	,	#								
1100					←	*	%	@								
1101					(	)	⌢	'								
1110					+	;	_	=								
1111					‡	¢	±	✓								

Figure 11. Extended Binary-Coded-Decimal Interchange Code

Bit Positions 76	00				01				10				11			
4321 \ X5	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL	DC_0			b̸ blank	0					@	P				P
0001	SOM	DC_1			!	1					A	Q			a	q
0010	EOA	DC_2			"	2					B	R			b	r
0011	EOM	DC_3			#	3					C	S			c	s
0100	EOT	DC_4 STOP			$	4					D	T			d	t
0101	WRU	ERR			%	5					E	U			e	u
0110	RU	SYNC			&	6					F	V			f	v
0111	BELL	LEM			'	7					G	W			g	w
1000	BKSP	S_0			(	8					H	X			h	x
1001	HT	S_1			)	9					I	Y			i	y
1010	LF	S_2			*	:					J	Z			j	z
1011	VT	S_3			÷	;					K	[			k	
1100	FF	S_4			,	<					L	\			l	
1101	CR	S_5			–	=					M	]			m	
1110	SO	S_6			.	>					N	↑			n	ESC
1111	SI	S_7			/	?					O	←			o	DEL

Figure 12. Eight-Bit Representation for American Standard Code for Information Interchange for Use in Eight-Bit Environment (ASCII-8)

ter-to-storage operation; SI, a storage and immediate-operand operation; and SS, a storage-to-storage operation.

For purposes of describing the execution of instructions, operands are designated as first and second operands and, in the case of BRANCH ON INDEX, third operands. These names refer to the manner in which the operands participate. The operand to which a field in an instruction format applies is generally denoted by the number following the code name of the field, for example, R_1, B_1, L_2, D_2.

In each format, the first instruction halfword consists of two parts. The first byte contains the operation code (op code). The length and format of an instruction are specified by the first two bits of the operation code.

INSTRUCTION LENGTH RECORDING

BIT POSITIONS (0-1)	INSTRUCTION LENGTH	INSTRUCTION FORMAT
00	One halfword	RR
01	Two halfwords	RX
10	Two halfwords	RS or SI
11	Three halfwords	SS

The second byte is used either as two 4-bit fields or as a single eight-bit field. This byte can contain the following information:

Four-bit operand register specification R_1, R_2, or R_3)
Four-bit index register specification (X_2)
Four-bit mask (M_1)
Four-bit operand length specification (L_1 or L_2)
Eight-bit operand length specification (L)
Eight-bit byte of immediate data (I_2)

In some instructions a four-bit field or the whole second byte of the first halfword is ignored.

The second and third halfwords always have the same format:

Four-bit base register designator (B_1 or B_2), followed by a 12-bit displacement (D_1 or D_2).

Address Generation

For addressing purposes, operands can be grouped in three classes: explicitly addressed operands in main storage, immediate operands placed as part of the in-

Appendix F. EBCDIC and ASCII-8 Charts

The numbering convention for the bit positions within a character differ for each of the codes. The conventions are:

	BIT POSITIONS
EBCDIC	01234567
ASCII-8	76X54321

Extended Binary-Coded-Decimal Interchange Code (EBCDIC)

Bit Positions → 01 → 23 → 4567

01 →	00				01				10				11			
4567 / 23 →	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL				ƀ blank	&	-						>	<	‡	0
0001							/		a	j			A	J		1
0010									b	k	s		B	K	S	2
0011									c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	EOB	UC					f	o	w		F	O	W	6
0111	DEL	IDL	PRE	EOT					g	p	x		G	P	X	7
1000									h	q	y		H	Q	Y	8
1001					.		,	"	i	r	z		I	R	Z	9
1010					?	!		:								
1011					.	$	,	#								
1100					←	*	%	@								
1101					(	)	⌢	'								
1110					+	;	_	=								
1111					‡	¢	÷	✓								

American Standard Code for Information Interchange (ASCII) Extended to Eight Bits

Bit Positions → 76 → X5 → 4321

76 →	00				01				10				11			
4321 / X5 →	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NULL	DC_0			ƀ blank	0					@	P				p
0001	SOM	DC_1			!	1					A	Q			a	q
0010	EOA	DC_2			"	2					B	R			b	r
0011	EOM	DC_3			#	3					C	S			c	s
0100	EOT	DC_4 STOP			$	4					D	T			d	t
0101	WRU	ERR			%	5					E	U			e	u
0110	RU	SYNC			&	6					F	V			f	v
0111	BELL	LEM			'	7					G	W			g	w
1000	BKSP	S_0			(	8					H	X			h	x
1001	HT	S_1			)	9					I	Y			i	y
1010	LF	S_2			*	:					J	Z			j	z
1011	VT	S_3			+	;					K	[			k	
1100	FF	S_4			,	<					L	\			l	
1101	CR	S_5			-	=					M	]			m	
1110	SO	S_6			.	>					N	↑			n	ESC
1111	SI	S_7			/	?					O	←			o	DEL

Appendix H-4

X3.2/139
June 10, 1964

RECEIVED BEMA/DPG
JUN 16 1964

Report of Meeting No. 12, ASA Task Group X3.2.4
Code Assignments
June 2 and 3, 1964
Princeton Inn, Princeton, New Jersey

5/63 #4 [illegible]

1. Attendance

Members and attendees	Affiliation	Status	June 2	June 3
J.L. Little, Acting Chairman	National Bureau of Stand.	M	X	X
O.R. Arne	Mergenthaler Linotype	M	X	X
J.F. Auwaerter	Teletype Corp.	M		
E.A. Avakian	Teleregister	M		X
J.B. Booth	Teletype Corp.	A	X	X
E.H. Clamons	Univac	A	X	X
C.J. Davis	IBM	M		
J.G. Doyle	Fairchild Graphic	M		
W.F. Emmons	IBM	A	X	X
R.F. Ewald	BTL	A		
E.S. Ferguson	Rockford Research Insti.	O	X	X
J.G. Griffis	DCA	M	X	X
R.M. Gryb	AT&T	M		
W.F. Huf	Navy	M	X	X
J.D. Keen	Univac Div.	M		
D.A. Kerr	Bell Telephone Labs.	A	X	X
W.Y. Lang	BTL	M	X	X
E.J. Lewis	ITT - DISD	M	X	X
D.V. Magee	USAEL Ft. Monmouth	O	X	
E.F. Manning	USAELRDL	M		
C.N. Mooers	Rockford Research Insti.	O	X	X
R.F. Mundy	GSA	O	X	X
J.K. Nelson	Western Union	M		
F.W. Smith	Western Union	A		
A.... Spaiholz	Fairchild Graphic	A		
A. Tholstrup	Friden	M		
.R. Turner	NASA	M	X	X
R.C. Umstead	USAELRDL	A		
E.F. Vidro, Jr.	Xerox Corp.	M	X	X
A.L. Whitman	ITT Communications Systems	M	X	X

M = Member
A = Alternate
O = Observer

2. Membership Status

No changes in membership were known. It was noted that Fairchild Graphic Systems, Western Union, Friden, AT&T, and USAELRDL were not represented during the two meetings.

3. Minutes of Previous Meeting

The minutes of the previous meeting were reviewed and accepted on the second day by a vote of 11 for, 0 against, and 0 abstentions.

4. Documents Assigned X3.2.4 Numbers

X3.2.4/

31	"Notes on Code Insensitivity in Digital Switching-II" by L.B. Haigh - ITT Communication Systems, Inc.
32	Letter to Task Group X3.2.4 by L.R. Turner, "Comments on Aspects of ISO/TC97/SC2 Actions on Code Assignment in 7-Bit Code Table".
33	"Brief Report of Meeting Held by TC97; 97/SC2; 97/SC4; 95/SC8; 53D in New York, May 1964" by J.F. Auwaerter.

5. Document ISO/TC97/SC2 (N.Y. '64-18) 123 *E Was Reviewed

This resulted in the following actions:

a) Mr. Kerr proposed that the spelling ENQ be accepted for inquiry for ISO use and that ASCII retain the INQ spelling. This was rejected by a vote of 2 for, 5 against, and 4 abstentions.

It was then proposed that the accepted spelling be ENQ for both ISO and ASCII and that WRU be changed to Enquiry. This was passed by a vote of 10 for, 0 against, and 1 abstention.

b) Mr. Kerr proposed, and Mr. Lewis seconded, that ACK be placed in position 0/6 of the code table replacing RU/CNCL. This passed by a vote of 8 for, 0 against, and 3 abstentions.

c) Mr. Kerr moved, and Mr. Griffis seconded, that the backspace symbol be made BS and be placed in position 0/8 which corresponds to FE_0. This passed by a vote of 10 for, 0 against, and 1 abstention.

d) Mr. Kerr proposed, and Mr. Clamons seconded, that the position 0/9 be changed from HT/SK to HT for both ISO and ASCII and that the definition for HT/SK given in X3.2/112 be retained. This passed by a vote of 11 for, 0 against, and 0 abstentions.

e) Mr. Clamons proposed, and Mr. Kerr seconded, that the position 0/11 V$_{tab}$ be replaced by VT in ASCII. This passed by a vote of 11 for, 0 against, and 0 abstentions.

f) Mr. Kerr proposed, and Mr. Lewis seconded, that the position 1/0, formerly known as TC$_7$ in the ISO Code Table and DC$_0$ in ASCII, be changed to DLE in both the ISO proposal and in ASCII. This was passed by a vote of 6 for, 2 against, and 2 abstentions.

g) Mr. Kerr proposed, and Mr. Lewis seconded, that LEM be removed from the Code Table and replaced by EM with the understanding that it is forward acting and should apply to both the ISO and ASCII Code Tables. Further, that the ISO definition be accepted with comment that it should be forward acting. This was accepted by a vote of 10 for, 1 against, and 0 abstentions.

h) Mr. Kerr proposed, and Mr. Turner seconded, that EOB be changed to ETB for both ISO and the ASCII Code Tables. This was accepted by a vote of 11 for, 0 against, and 0 abstentions.

i) Mr. Lewis proposed, and Mr. Kerr seconded, that ETB be placed in position 1/7 and EM in position 1/9 for both the ISO and ASCII Code Tables. This was accepted by a vote of 9 for, 1 against, and 1 abstention.

j) Mr. Kerr proposed, and Mr. Lewis seconded that CNCL be placed in position 1/8 in both the Code Tables for ISO and ASCII. This passed with a vote of 8 for, 0 against, and 3 abstentions.

k) Mr. Lewis proposed, and Mr. Kerr seconded that ESC be placed in position 1/11 for ISO. This passed with a vote of 7 for, 1 against, and 3 abstentions.

l) Mr. Lang proposed, and Mr. Lewis seconded, that SS be placed in position 1/10 with the comment that some means is required to end the special sequence which, for example, might be by the use of SI for the ISO. This passed with a vote of 12 for, 0 against, and 0 abstentions. This was extended to cover ASCII also when proposed by Mr. Kerr and seconded by Mr. Emmons. This passed by a vote of 12 for, 0 against, and 0 abstentions.

m) Mr. Kerr proposed, and Mr. Griffis seconded, that the position 1/11 be used for ESCAPE in ASCII. This was passed by a vote of 10 for, 1 against, and 1 abstention.

n) Mr. Kerr proposed, and Mr. Lewis seconded, that position 2/0 be called SPACE instead of a word separator for ISO with the suggestion that the approved abbreviation be SP and that its abbreviation be SP instead of a B with a slash through the top for ASCII. This passed with a vote of 12 for, 0 against, and 0 abstentions.

o) Mr. Lang proposed, and Mr. Turner seconded, that the underline be moved to the position 4/0 and that the @ with footnote (3) be placed in position 6/0 for both ISO and the ASCII Code Tables. This passed with a vote of 12 for, 0 against, and 0 abstentions.

p) Mr. Kerr proposed, and Mr. Turner seconded, that position 5/12 be designated a CS$_2$ with footnote (3), position 5/14 be a tilde with footnotes (7) (8) and position 5/15 be accent grave for ISO and for ASCII. This was passed by a vote of 9 for, 2 against, and 1 abstention.

q) Mr. Booth proposed, and Mr. Kerr seconded, that position 5/11 be a left bracket and 5/12 second currency and 5/13 be the right bracket , 6/0 be an @ symbol, 7/11, 7/12, 7/13, 7/14 respectively, be reserved for national graphics usage with footnote (3) and position 2/4 CS$_1$ with footnote (5). This passed by a vote of 11 for, 1 against, and 0 abstentions.

The definitions that were contained in Document 1, 2, 3 *E then discussed:

A. Null It was felt that the definition contained in this ISO paper was too limited and, therefore, the following comment should be returned to the ISO group. "The potential use of this character for such purposes as advancing but not the punching of record medium is in conflict with the ISO proposed definition. Any reference to the ability of some forms of transmission systems to accept this character are more appropriately the subject of future standards relating to the transmission of the six and seven bit codes. We suggest that the following is more general:

The all zero character which may serve to accomplish a) time fill, and b) media fill." This comment was proposed by Mr. Kerr and seconded by Mr. Booth. This resolution passed by a vote of 12 for, 0 against, and 0 abstentions.

B. Mr. Kerr proposed, and Mr. Turner seconded, that the definition for SOH be accepted without comment. This passed by a vote of 12 for, 0 against, and 0 abstentions.

C. Mr. Lang proposed, and Mr. Lewis seconded, that the definition for STX be accepted for ISO and should apply also to ASCII. This was passed by a vote of 8 for, 3 against, and 1 abstention.

D. Other comments on the definitions contained in the ISO document reviewed are summarized below:

ACK - Questioned the validity of the word "correct" in part (a) of the definition.

Format Effectors - These should reflect usage not only in printing but also in displays.

DLE and ESC - The words "next single code combination" is felt to be misleading and obscure, since sequence of bit patterns following these characters may have meanings beyond their usual meanings in the code table.

EM - Is understood to be a "forward acting" control function.

EOT - Remove the statement "(everyone of which must be terminated by ETX)".

ETB - Delete the second sentence, "This character may be added to or deleted from a character sequence at any point in the data flow path without altering the interpretation of data.", as being too broad and general. System studies must be made to clarify its use.

IS - In the definition of group separator, delete the third "it" and place a comma after the first "includes".

In the definition of file separator add a comma after the first "includes".

SI - In ASCII this may not carry such a strong connotation of "restore", but this has not yet been established.

SO - In ASCII this may be restricted to graphic characters only, but this has not yet been established.

SS - Delete the second sentence, "This SS may be used for transmission control purposes.", to retain a clear separation between "end to end" and "link" control characters.

The remainder of the definitions in the document were accepted without change.

E.F. Vidro, Jr.

E.F. Vidro, Jr.
Acting Secretary, X3.2.4

REPORT TO ASA SUBCOMMITTEE X3.2

From Task Group X3.2.4

5/30 4+ MM

As a result of actions taken by Task Group X3.2.4 at Princeton, New Jersey on June 2 and 3, 1964, it is recommended that Subcommittee X3.2 approve the following items relevant to "A Third Draft Proposal on ISO 6 Bit and 7 Bit Codes", as given in Document ISO/TC97/SC2 (N.Y. '64-18)123* E, dated May 20, 1964. Specifically, relating to the ISO 7 Bit Code, X3.2 should:

1. Approve all of the code designations and position assignments given in columns 0 and 1.
2. Approve the assignment of UNDERLINE in position 4/0.
3. Approve the assignment of "@" in position 6/0.
4. Approve the assignment of Footnote 3 in the eight positions 5/11, 5/12, 5/13, 6/0, 7/11, 7/12, 7/13, and 7/14 as "Reserved for National Use" graphics.
5. Approve the assignment of Footnote 5 in position 2/4 as a "Reserved for National Use" currency sign.
6. Approve the assignment of Footnote 8 in the three positions 2/2, 2/7, and 5/14 (which provides for a change in the meaning of these symbols to diacriticals when they follow the Backspace code).
7. Approve the assignment of Footnote 7 in position 5/14 to allow for an optional diacritical sign (tilde) in certain countries.
8. Accept the ISO definitions for the specific controls as given in paragraph 3.2 of Document 123*E, but with the following comments:

   (a) ACK - We question the validity of the word "correct" in part (a) of the definition.

   (b) All of the Format Effectors should reflect a usage not only in printing, as given, but also in displays.

   (c) DLE and ESC - The words "next single following code combination" appear to be misleading and obscure, since sequences of bit patterns following these characters may have meanings changed from their usual meanings in the code table.

   (d) EM - This is understood to be a "forward acting" control function.

   (e) EOT - The parenthetical statement should be removed, since it appears at this time to be unduly restrictive.

   (f) ETB - The second sentence should be removed. That sentence is too broad and general. Further system studies are required to clarify such use. The sentence also suggests to the user, falsely, that he can insert ETB characters.

   (g) IS - Under "group", delete the third "it". Under "group" and "file", add a comma after the first "includes".

   (h) NULL - The potential use of this character for such purposes as advancing but not punching of a record medium is in conflict with the ISO proposed definition. Any reference to the ability of some forms of transmission systems to accept this character are more appropriately the subject of future standards relating to the transmission of the 6 and 7 bit codes. Until such time, we recommend a less restrictive definition, such as:

      "The all zeros character which may serve to accomplish:

      a) Time fill,
      b) Media fill."

   (i) SI - In ASCII this may not carry such a strong connotation of "Restore", but this is not yet established.

   (j) SO - In ASCII this may be restricted to graphic characters only, but this is not yet established.

   (k) Space - An abbreviation to be shown in the code table in place of "Space" should be "Sp".

   (l) SS - Remove the second sentence, in order to maintain a clear separation between "end-to-end" and "link" control characters.

With respect to ASCII, it appears that X3.2.4 will recommend, at a later time, that the 32 control functions in columns 0 and 1 be brought into exact agreement with the ISO 7-bit Code Table, except for details relating to the symbols enclosed in parentheses, and the exact use of the parentheses marks. For the time being, the control definitions given in X3.2/112 are preferred for ASCII, but

Data Processing Group

bema

Business Equipment Manufacturers Association
235 East 42nd Street, New York, N.Y. 10017 – MUrray Hill 7-5969

X3.2/139
June 10, 1964

MINUTES OF MEETING #38 - SUBCOMMITTEE X3.2

Date: June 4, 1964

Presiding Officer: L. L. Griffin - Chairman, Subcommittee X3.2

Time: 8:30 A.M. to 4:00 P.M. - with recess for lunch.

Place: Princeton Inn, Princeton, New Jersey

1. Attendance (See Appendix A)

2. Minutes of previous meetings

No comments.
Postponed any action until the next meeting.

RECEIVED BEMA/DPG
JUN 17 1964

3. Membership

3.1 No change in membership.

3.2 Mr. D. A. Kerr was given a proxy for Mr. R. M. Gryb.

3.3 Mr. C. E. Schultheiss was given a proxy for Mr. C. E. Macon.

3.4 Mr. E. A. Avakian gave proxy to Mr. E. S. Fergusson for the afternoon session.

3.5 The Chairman announced Mr. J. E. Taunt has accepted Chairmanship of X3.2.5.

3.6 The Chairman announced Mr. J. L. Little has accepted Chairmanship of X3.2.4.

NMA+ 3/8/5

X3.2 - June 4, 1964
(Appendix A)

X3.2/139
June 10, 1964

Name	Company	Status
C. E. Schultheiss	Burroughs	A
R. H. Brown	DOD	M
H. S. White, Jr.	DOD	A
E. J. Lewis	ITT - DISD	M
O. R. Arne	Mergenthaler Linotype Co.	M
R. E. Mullendore	Census	A
V. L. Thompson	UNIVAC	A
T. R. Bousquet	Honeywell	M
E. A. Avakian	Teleregister	M
E. F. Vidro, Jr.	Xerox, Corp.	M
G. V. Hawkins	IBM	A
W. J. Craven	NAVCOSSACT	O
H. L. Gross	NCR	A
J. W. Sage	U. S. Army	O
W. Y. Lang	Bell Telephone Laboratories	M
R. F. Ewald	Bell Telephone Laboratories	A
F. W. Smith	Western Union Tel. Co.	A
R. F. Mundy	GSA	O
J. F. Auwaerter	Teletype	M
A. L. Whitman	ITT Communications Systems	M
H. J. Smith, Jr.	IBM; ACM Std's Comm	C
C. E. Mackenzie	IBM	M
L. L. Griffin	U. S. Navy	M
J. L. Little	NBS	M
J. E. Taunt	GE	M
G. L. Murphy	Ampex	M
L. R. Turner	NASA	O
W. F. Huf	U. S. Navy	O
D. A. Kerr	Bell Telephone Laboratories	A
J. B. Booth	Teletype	A
A. E. Spencer, Jr.	Bell Telephone Laboratories	O
W. H. McKenzie	ITT	O
C. N. Mooers	Rockford Research	O
E. S. Fergusson	Rockford Research	O
E. H. Clamons	UNIVAC	M
A. H. Hassan	NCR	M
H. Spielman	RCA	M

July 1984

affilia
w
193

X3.2/161
October 12, 1964

TO: Members of X3.2 and Task Groups

FROM: Eric H. Clamons

SUBJECT: ISO/TC97/SC2(Secr. 32) 125 F/E Draft Proposal

Attached to this letter is the ISO/TC97/SC2(Secr. 32) 125 F/E Draft Proposal. With the exception of pages 6 and 8 only the odd pages have been reproduced since the others contain the French version of the standard.

This document has not yet been received officially by the ASA and is made available to X3.2 for reference only.

Eric H Clamons
Eric H. Clamons

EHC:fbk
Enc.

CBI 67/3/4

Secrétariat : **France**

OCT 6 1964
[illegible]

ISO/TC 97/SC 2 (Secr. 32) 125 F/E

Juillet 1964

ISO

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION

Comité technique 97 : CALCULATEURS ET TRAITEMENT DE L'INFORMATION

Sous-Comité 2 : « JEUX DE CARACTÈRES ET CODAGE »

TROISIÈME AVANT-PROJET

CARACTÈRES CODÉS A 6 ET 7 ÉLÉMENTS POUR L'ÉCHANGE D'INFORMATION ENTRE MATÉRIELS DE TRAITEMENT DE L'INFORMATION

THIRD ISO DRAFT PROPOSAL

6 AND 7 BIT CODED CHARACTER SETS FOR INFORMATION PROCESSING INTERCHANGE

AFNOR

Association Française de Normalisation — 23, rue Notre-Dame-des-Victoires — Paris-2e

BRIEF HISTORY

The present Recommendation was prepared by Technical Committee ISO/TC 97, Subcommittee 2, the Secretariat of which is held by AFNOR.

The Technical Committee began its work in 1962; national contributions were submitted at the Paris May 1962 meeting by the following P members of Subcommittee 2 : France, Germany, Italy, United Kingdom, United States.

A synthesis of the various proposals was achieved and the resulting draft circulated for further discussions in October 1962.

After that meeting, Committee ISO/TC 97 circulated, for vote and comments, the first draft proposal [as contained in document ISO/TC 97/SC 2 (Secr. 12) 30 F/E, January 1963]. A majority (7 for, 3 against, and 2 not replying) was in favor of this draft proposal [see document ISO/TC 97/SC 2 (Secr. 14) 37].

During its October 1963 meeting, Subcommittee 2 considered the various comments received, and a second draft proposal [ISO/TC 97/SC 2 (Secr. 22) 75 F/E] embodied all the agreed amendments. Following Resolution N° 1 of this meeting, the second draft proposal was circulated for vote and comments among (P) and (O) members of ISO/TC 97.

The result of this second letter ballot [as shown in document ISO/TC 97/SC 2 (Secr. 24) 77] was 11 for approval (Belgium, France, Germany, Italy, Japan, Poland, Spain, Sweden, Switzerland, United Kingdom, United States), none against. However, as most of the comments were of substance, document 75 was reconsidered at the New York meeting of ISO/TC 97/SC 2 (May, 11-15 1964).

The third draft proposal embodies all the amendments agreed upon during the May 1964 meeting.

Following Resolution N° 6 of the meeting, the present document is to be submitted to (P) members of ISO/TC 97 for vote, and to (O) members and liaison organizations for information or possible comments.

I. — INTRODUCTION

SUBJECT

1. This Recommendation defines two character sets for general usage, including letters, figures, punctuation marks, other symbols and controls, with their coded representation.

2. The requirements for graphics and controls in data handling and programming, in accordance with computer and ancillary equipment characteristics, have been taken into account in determining these character sets.

3. The Recommendation consists of two tables with notes, a legend and explanatory notes.
Functional characteristics related to control characters are described in an Appendix, which is not a part of the Recommendation.
A possibly revised description of those functional characteristics should become part of the Recommendation when more experience has been gained with the use of the character sets.

SCOPE

1. These character sets are primarily intended for the interchange of information among data processing systems and associated equipment. They may be regarded as basic alphabets in the abstract sense.

2. These character sets are applicable to all so-called latin alphabets.

3. One character set contains 6-bit characters and provides 64 coded characters, the other contains 7-bit characters and provides 128 coded characters. The two sets are compatible in that they are logically related. This facilitates the conversion between the two sets.

IMPLEMENTATION

1. These character sets may be implemented in various media with logically related codes, as appropriate. For example, this could include punched tapes, punched cards, magnetic tapes and transmission channels, thus permitting interchange of data to take place either indirectly by means of an intermediate recording in a physical medium, or by local electrical connection of various units (such as input and output devices and computers) or by means of data transmission equipment.

2. It is the subject of other ISO Recommendations to describe the means of recording these character sets in physical media and the means of encoding them for transmission while taking into account the need for error checking (*).

(*) For implementation of codes in punched tape, see Recommendation ISO/R... [At the present time, draft ISO proposal ISO/TC 97/SC 2 (Sec. 33) 126, July 1964].

II. TABLEAUX ET NOTES
TABLES AND NOTES

Caractères codés pour l'échange d'information entre matériels de traitement de l'information

Coded character sets for information processing interchange.

TABLEAU DE CODE A 6 ÉLÉMENTS D'INFORMATION

6 BIT SET TABLE

				b_6 →	0	0	1	1
				b_5 →	0	1	0	1
Bits b_4	b_3	b_2	b_1	Column / Row	0	1	2	3
0	0	0	0	0	F_0 Space	0	Null	P
0	0	0	1	1	F_1 (HT)	1	A	Q
0	0	1	0	2	F_2 (LF) ①	2	B	R
0	0	1	1	3	F_3 (VT)	3	C	S
0	1	0	0	4	F_4 (FF)	4	D	T
0	1	0	1	5	F_5 (CR) ①	5	E	U
0	1	1	0	6	SO	6	F	V
0	1	1	1	7	SI	7	G	W
1	0	0	0	8	(	8	H	X
1	0	0	1	9	)	9	I	Y
1	0	1	0	10	*	: ②	J	Z
1	0	1	1	11	+	; ②	K	([) ③
1	1	0	0	12	,	< ⑤ CS$_1$ ④	L	(CS$_2$) ④ ③
1	1	0	1	13	−	= %	M	(]) ③
1	1	1	0	14	.	> &	N	Esc
1	1	1	1	15	/	´	O	Delete

Juillet 1964

July 1964

NOTES RELATIVES

AU TABLEAU DE CODE A 6 ÉLÉMENTS D'INFORMATION

① Les commandes CR et LF sont destinées aux appareils d'impression qui requièrent des combinaisons distinctes pour le retour de chariot et l'interligne.
Pour les appareils ne comportant qu'une seule commande pour l'opération combinée de retour de chariot et d'interligne, la fonction F_2 signifiera « Retour à la ligne » (NL). Ceci exige un accord entre l'émetteur des données et leur destinataire. F_5 signifiera alors « Retour arrière » (BS).

② S'il est nécessaire de représenter chacun des nombres 10 et 11 par un caractère unique (par exemple, pour les subdivisions de la Livre Sterling), ces nombres devront prendre respectivement les places de « deux points » (:) et de « point virgule » (;). Ces substitutions exigent un accord entre l'émetteur des données et leur destinataire.

③ « Réservé pour usage national ». Ces positions sont prévues principalement pour des extentions alphabétiques. Si elles ne sont pas requises pour de telles extensions, on peut les utiliser pour des symboles, et le choix recommandé est indiqué entre parenthèses.

④ La représentation graphique et la signification de chacun des deux symboles monétaires CS_1 (position 1/12) et CS_2 (position 3/12) peuvent varier en fonction des usages nationaux.

⑤ Il y a lieu de choisir l'un ou l'autre des deux groupes de trois symboles qui sont représentés dans ces positions : l'interprétation des combinaisons correspondantes exige un accord entre l'émetteur des données et leur destinataire.

NOTES ABOUT

THE 6 BIT SET TABLE

① The controls CR and LF are intended for printer equipment which requires separate combinations to return the carriage and to feed a line.
For equipment which uses a single control for a combined carriage return and line feed operation, the function F_2 will have the meaning of « New Line » (NL). This requires agreement between the sender and the recipient of the data. F_5 will then have the meaning of « Backspace » (BS).

② If 10 and 11 as single characters are needed (for example, for Sterling currency subdivision), they should take the place of « colon » (:) and « semi-colon » (;) respectively. These substitutions require agreement between the sender and the recipient of the data.

③ « Reserved for National Use ». These positions are primarily intended for alphabetic extensions. If they are not required for that purpose they may be used for symbols and the recommended choice is shown in parentheses.

④ The graphical representation and the meaning of each of the two currency signs CS_1 (position 1/12) and CS_2 (position 3/12) may vary according to national use.

⑤ Either of the two sets of three symbols shown in these positions in the table may be chosen; the interpretation of the corresponding combinations requires agreement between the sender and the recipient of the data.

Caractères codés pour l'échange d'information entre matériels de traitement de l'information.

TABLEAU DE CODE A 7 ÉLÉMENTS D'INFORMATION

Coded character sets for information processing interchange.

7 BIT SET TABLE

Bits				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1	Row / Column	0	1	2	3	4	5	6	7
0	0	0	0	0	(TC_0) Null	(TC_7) DLE	Space	0	_	P	(@) ③	p
0	0	0	1	1	(TC_1) SOH	DC_1	!	1	A	Q	a	q
0	0	1	0	2	(TC_2) STX	DC_2	" ⑧	2	B	R	b	r
0	0	1	1	3	(TC_3) ETX	DC_3	# ⑥	3	C	S	c	s
0	1	0	0	4	(TC_4) EOT	DC_4 (Stop)	CS_1 ④	4	D	T	d	t
0	1	0	1	5	(TC_5) Enq	(TC_8) Nack	%	5	E	U	e	u
0	1	1	0	6	(TC_6) Ack	(TC_9) Sync	&	6	F	V	f	v
0	1	1	1	7	Bell	(TC_{10}) ETB	' ⑧	7	G	W	g	w
1	0	0	0	8	FE_0 (BS)	Cncl	(	8	H	X	h	x
1	0	0	1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE_2 (LF) ①	SS	*	: ②	J	Z	j	z
1	0	1	1	11	FE_3 (VT)	Esc	+	; ②	K	([) ③	k	③
1	1	0	0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	(CS_2) ④ ③	l	③
1	1	0	1	13	FE_5 (CR) ①	IS_3 (GS)	-	=	M	(]) ③	m	③
1	1	1	0	14	SO	IS_2 (RS)	.	>	N	^ ⑦ ⑧	n	③
1	1	1	1	15	SI	IS_1 (US)	/	?	O	`	o	Delete

Juillet 1964
July 1964

NOTES RELATIVES AU TABLEAU DE CODE A 7 ÉLÉMENTS D'INFORMATION

① Les commandes CR et LF sont destinées aux appareils d'impression qui requièrent des combinaisons distinctes pour le retour de chariot et l'interligne.
Pour les appareils ne comportant qu'une seule commande pour l'opération combinée de retour de chariot et d'interligne, la fonction FE_2 signifiera « Retour à la ligne » (NL). Ceci exige un accord entre l'émetteur des données et leur destinataire.

② S'il est nécessaire de représenter chacun des nombres 10 et 11 par un caractère unique (par exemple, pour les subdivisions de la Livre Sterling), ils devront prendre respectivement les places de « deux points » (:) et de « point virgule » (;). Ces substitutions exigent l'accord entre l'émetteur des données et leur destinataire.

③ « Réservé pour usage national ». Ces positions sont prévues principalement pour des extensions alphabétiques. Si elles ne sont pas requises pour de telles extensions, on peut les utiliser pour des symboles et un choix recommandé est indiqué entre parenthèses dans certains cas.

④ La représentation graphique et la signification de chacun des deux symboles monétaires CS_1 (position 2/4) et CS_2 (position 5/12) peuvent varier en fonction des usages nationaux.

⑥ Le signe « numéro » (#), à la position 2/3, peut avoir une seconde représentation graphique (N^{o}).

⑦ La représentation du « tilde » (~) par l'« accent circonflexe » (^) est admissible pour les échanges internationaux d'information en espagnol et en portugais. Dans les pays de langues espagnole et portugaise, le « tilde » pourra remplacer « l'accent circonflexe » à la position 5/14.

⑧ Les symboles graphiques qui figurent aux positions 2/2, 2/7, 5/14 signifient respectivement « guillemet », « apostrophe » et « flèche en haut »; cependant, ces caractères prennent la signification des signes diacritiques « tréma », « accent aigu » et « accent circonflexe » lorsqu'ils viennent à la suite d'un caractère « Retour arrière ».
Le symbole en position 5/15 n'a pas reçu de dénomination autre que celle d'« accent grave » qui correspond à son usage en tant que signe diacritique.

NOTES ABOUT THE 7 BIT SET TABLE

① The controls CR and LF are intended for printer equipment which requires separate combinations to return the carriage and to feed a line.
For equipment which uses a single control for combined carriage return and line feed operation, the function FE_2 will have the meaning of « New Line» (NL). This requires agreement between the sender and the recipient of the data.

② If 10 and 11 as single characters are needed (for example, for Sterling currency subdivision), they should take the place of « colon » (:) and « semi-colon » (;) respectively. These substitutions require agreement between the sender and the recipient of the data.

③ « Reserved for National Use ». These positions are primarily intended for alphabetic extensions. If they are not required for that purpose, they may be used for symbols and a recommended choice is shown in parentheses in somecases.

④ The graphical representation and the meaning of the two currency signs CS_1 (position 2/4) and CS_2 (position 5/12) may vary according to national use.

⑥ The « number sign » (#) in position 2/3 may have an alternate graphical representation (N^{o}).

⑦ It is acceptable to represent « tilde » (~) by « circumflex accent » (^) for international interchange of information in Spanish and Portuguese languages. In Spanish and Portuguese speaking countries, the « tilde » may replace the « circumflex accent » in position 5/14.

⑧ The graphics in positions 2/2, 2/7, 5/14 have respectively the significance of « quotation mark », « apostrophe » and « upwards arrow »; however, these characters take on the significance of « diaeresis », « acute accent » and « circumflex accent » diacritical signs when they follow the « Backspace » character.
The symbol shown in position 5/15 did not receive any denomination other than « grave accent » which corresponds to its use as a diacritical sign.

III. — LEGEND

I. CONTROLS

Ack	Acknowledge
Bell	Bell
BS	Backspace
Cncl	Cancel
CR	Carriage Return
DC	Device Control
Delete	Delete
DLE	Data Link Escape
EM	End of Medium
Enq	Enquiry
EOT	End of Transmission
Esc	Escape
ETB	End of Transmission Block
ETX	End of Text
F	Function
FE	Format Effector
FF	Form Feed
FS	File Separator
GS	Group Separator
HT	Horizontal Tabulation
IS	Information Separator
LF	Line Feed
Nack	Negative Acknowledge
NL	New Line
Null	Null
RS	Record Separator
SI	Shift-in
SO	Shift-out
SOH	Start of Heading
Space	Space
SS	Start of Special Sequence
Stop	Stop
STX	Start of Text
Sync	Synchronous Idle
TC	Transmission Control
US	Unit Separator
VT	Vertical Tabulation

2. GRAPHICAL SYMBOLS

Graphical Representation	Name	Position in the code tables	
		6-bit	7-bit
(Space)	A normally non-printing graphic character	0/0	2/0
!	Exclamation mark	—	2/1
"	Quotation mark, Diaeresis (note ⑧)	—	2/2
# Nº	Number sign (note ⑥)	—	2/3
(CS_1)	First currency symbol (note ④)	1/12	2/4
%	Percent	1/13	2/5
&	Ampersand	1/14	2/6
'	Apostrophe, Acute accent (note ⑧)	1/15	2/7
(	Left parenthesis	0/8	2/8
)	Right parenthesis	0/9	2/9
*	Asterisk	0/10	2/10
+	Plus sign	0/11	2/11
,	Comma	0/12	2/12
-	Hyphen, minus sign	0/13	2/13
.	Full stop (period)	0/14	2/14
/	Solidus	0/15	2/15
:	Colon	1/10	3/10
;	Semi-colon	1/11	3/11
<	Less than	1/12	3/12
=	Equals	1/13	3/13
>	Greater than	1/14	3/14
?	Question mark	—	3/15
_	Underline	—	4/0
[	Left square bracket	3/11	5/11
(CS_2)	Second currency symbol (note ④)	3/12	5/12
]	Right square bracket	3/13	5/13
^	Upwards arrow, Circumflex accent (note ⑧)	—	5/14
`	Grave accent (note ⑧)	—	5/15
@	Commercial at	—	6/0

Note ④ See pages 7 and 9

Note ⑥ See page 9

Note ⑧ See page 9

IV. — EXPLANATORY NOTES

1. NUMBERING OF THE POSITIONS IN THE CODE TABLES

Within any one character the bits are identified by b_7, b_6...b_1, where b_7 or b_6 is the highest order, or most significant bit, and b_1 is the lowest order, or least significant bit.

If desired these may be given a numerical significance in the binary system, thus:

Bit identification:	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Significance :	64	32	16	8	4	2	1

In the code tables the columns and rows are identified by numbers written in binary and decimal notations.

Any one position in a code table may be identified either by its bit pattern, or by its column and row numbers. For instance, the position containing the figure 1 in the 7-bit set table may be identified :

— by its bit-pattern, e.g. 011 0001

— by its column and row numbers, e.g. 3/1

2. DIACRITICAL SIGNS

In the 7 bit character set, some printing symbols may be designed to permit their use for the composition of accented letters when necessary for general interchange of information. A sequence of three characters, comprising a letter, «Backspace» and one of these symbols, is needed for this composition, and the symbol is then regarded as a diacritical sign. It should be noted that these symbols take on their diacritical significance only when they follow the « Backspace» character; for example, the symbol corresponding to the code combination 2/7 (′) normally has the significance of « apostrophe», but becomes the diacritical sign « acute accent» when preceded by the «Backspace» character.

In order to increase efficiency, it is possible to introduce accented letters (as single characters) in the positions marked by note ③ in the code tables. According to national requirements, these positions may contain special diacritical signs, such as cedilla (¸), háček (ˇ), etc.

APPENDIX

FUNCTIONAL CHARACTERISTICS RELATED TO CONTROL CHARACTERS

(This Appendix is not a part of the Recommendation)

1. GENERALIZED CONTROLS

Generalized controls are designated by a specific name followed by a subscript number.

They are defined as follows :

TC — **Transmission Control** — A functional character intended to control or facilitate transmission of information over telecommunication networks.

FE — **Format Effector** — A functional character which controls the lay-out or positioning of information in Input/Output media.

DC — **Device Control** — A functional character for the control of an ancillary device associated with a data processing or telecommunication system, more especially switching a device « On » or « Off ».

An example of how device control characters could be used in a specific tape system employing two ancillary tape punches and one ancillary tape reader is as follows :

DC_1 — First Punch « On »
DC_2 — Second Punch « On »
DC_3 — Tape reader « On »

IS — **Information Separator** — A character which is used to separate and qualify information in a logical sense. There is a group of four such characters, which are intended to be used in a hierarchical order.

F — **Function** — A functional character in the 6-bit set, which may have the combined or alternative meanings of IS or FE.

Most generalized controls TC, FE, DC, IS, and F, have specific meanings which are shown next to them, in parentheses, in the code tables. Their names, which are listed in paragraph 2, have been chosen from the terminology used for telecommunication and character-by-character printer equipment. In other types of equipment, different but related meanings may be associated with certain of the generalized controls, but this requires agreement between the sender and the recipient of the data.

2. SPECIFIC CONTROLS

In the tables, these controls are specifically designated by a name or an abbreviation.

They appear either alone (e.g. SO, SI), or in parentheses, associated with a generalized control character.

They are defined as follows :

Ack — **Acknowledge** — A Transmission Control character that is generated at a receiving device to indicate to the sending device that :

(a) A correct connection has been established,
or (b) Data has been received successfully.

Bell — **Bell** — A character for use when there is a need to call for human attention; it may control alarm or attention devices.

BS — **Backspace** — A layout character which controls the movement of the printing position one printing space backward on the same printing line.

Cncl — **Cancel** — A character used to indicate that the data with which it is associated is in error or is to be disregarded.

CR — **Carriage Return** — A layout character which controls the movement of the printing position to the first printing position on the same printing line.

Delete — **Delete** — This character is used primarily to erase or obliterate erroneous or unwanted characters in punched tape.

DLE — **Data Link Escape** — A Transmission Control character which will change the meaning of the next single following code combination (see « Escape »). It is used exclusively to provide supplementary controls in data transmission networks.

EM — **End of Medium** — A control character which may be used to identify the physical end of the medium, or the end of the used, or wanted, portion of information recorded on a medium. The position of this character does not necessarily correspond to the physical end of the medium.

Enq — **Enquiry** — A Transmission Control character reserved for use in data communication systems as a request for a response from a remote station. The response may include station identification and/or station status.

EOT — **End of Transmission** — A Transmission Control character used to indicate the conclusion of the transmission of one or more texts (every one of which must be terminated by the ETX character).

Esc — **Escape** — A functional character which may be used for an extension of the standard character set of the code by changing the meaning of the next single following code combination. The precise meaning of this following character requires prior agreement between the sender and the recipient of the data. The « Escape» character itself may therefore be regarded as a warning or non-locking shift character.

The single character which follows « Escape» may be interpreted as a graphic or control character which is not included in the standard set.

The meaning of the single character following « Escape » may be « Go into Code X and stay in it ». In such a case, the sequence of two characters has the effect of causing a locking shift to occur.

ETB — **End of Transmission Block** — A Transmission Control character used to indicate the end of a block of data for transmission purposes. This character may be added to or deleted from a character sequence at any point in the data flow path without altering the interpretation of data. ETB is used for blocking data where the block structure has no necessary relationship with the processing format.

ETX — **End of Text** — A Transmission Control character used to terminate a sequence of characters started with an STX character and transmitted as an entity. In systems employing error-control, there is a possibility that the error-control character(s) may follow the ETX character.

FF — **Form Feed** — A layout character which controls the movement of the printing position to the first predetermined printing line on the next form.

FS — **File Separator** (see US for definition).

GS — **Group Separator** (see US for definition)

HT — **Horizontal Tabulation** — A layout character which controls the movement of the printing position to the next in a series of predetermined positions along the printing line.

LF — **Line Feed.** — A layout character which controls the movement of the printing position to the next printing line.

Nack — **Negative Acknowledge** — A Transmission Control character transmitted by a receiver as a negative response to the sender.

NL — **New Line** — A layout character which controls the movement of the printing position to the first printing position of the next printing line.

Null	— **Null** — A character which may be inserted into or removed from a stream of data without affecting the information content of that stream. Some forms of transmission systems may not be able to accept the use of this character.
RS	— **Record Separator** (see US for definition).
SI	— **Shift-in** — The shift-in character means that the code combinations which follow shall be interpreted according to the standard code table. If a special sequence contains a SO character, the SI character, which terminates the SO mode, may also terminate the special sequence (see below SS). The special sequence may then, if required, be immediately re-initiated by using the SS character.
SO	— **Shift-out** — The shift-out character means that the code combinations which follow shall be interpreted as outside of the standard code table until a shift-in character is reached.
SOH	— **Start of Heading** — A Transmission Control character used at the beginning of a sequence of characters which comprise a machine-sensible address or routing information. Such a sequence is referred to as the « Heading». An STX character has the effect of terminating a heading sequence.
Space	— **Space** — A normally non-printing graphic character used to separate words. It is also a layout character which controls the movement of the printing position, one printing position forwards.
SS	— **Start of Special Sequence** — A character which is used to indicate the start of a variable length sequence of characters which have special significance or which are to receive special handling. The SS character may be used for transmission control purposes. If a character is required for terminating a special sequence, the SI character should be used.
Stop	— **Stop** — This device control character is used to interrupt or turn off ancillary devices.
STX	— **Start of Text** — A Transmission Control character which precedes a sequence of characters that is to be treated as an entity and entirely transmitted through to the ultimate destination. Such a sequence is referred to as «Text». The STX character may be used to terminate a sequence of characters started by a SOH character.
Sync	— **Synchronous Idle** — A Transmission Control character used by a synchronous transmission system in the absence of any other character (idle condition) to provide a signal from which synchronism may be achieved or retained.
US	— **Unit Separator — Unit Separator (US), Record Separator (RS), Group Separator (GS) ; File Separator (FS),** are information separators which have a hierarchical relationship (in the order above) with each other, where the unit separator (US) is the least powerful of the set. The information separators follow the path to which they apply. A **« unit »** cannot include a «file», «group», or «record». A **« record »** may include (and if it includes, it includes completely) a variable number (none, one or more) of « units». A « record » cannot include a «file» or «group». A **« group »** may include (and if it includes, it includes completely) a variable number (none, one or more) of «records» and/or «units». A «group» cannot include a «file». A **« file »** may include (and if it includes, it includes completely) a variable number (none, one or more) of «groups», «records», and/or «units».
VT	— **Vertical Tabulation** — A layout character which controls the movement of the printing position to the next in a series of predetermined printing lines.

NMAH 316/5

Appendix F

Report of Meeting No. 13, ASA Task Group X3.2.4
Code Assignments
July 14 and 15, 1964
Hotel Leamington, Minneapolis, Minnesota

1. Attendance

Members and Attendees	Affiliation	Status	July 14	July 15
J.L. Little, Chairman	National Bureau of Stand.	M	X	X
O.R. Arne	Mergenthaler Linotype	M	X	X
J.F. Auwaerter	Teletype Corp.	M		
E.A. Avakian	Teleregister	M	X	X
J.B. Booth	Teletype Corp.	A	X	X
E.H. Clamons	Univac	A	X	X
H.D. Davids	Bell Telephone Co. of Canada	O	X	
C.J. Davis	IBM	M	X	X
W.F. Emmons	IBM	A	X	X
R.F. Ewald	BTL	A	X	
E.S. Ferguson	Rockford Research Insti.	O	X	X
J.G. Griffis	DCA	M	X	X
R.M. Gryb	AT&T	M		
W.F. Huf	Navy	M	X	X
J.D. Keen	Univac Div.	M		
D.A. Kerr	Bell Telephone Labs	A	X	X
W.Y. Lang	BTL	M		
E.J. Lewis	ITT - DISD	M		
B.V. Magee	USAEL Ft. Monmouth	M		
R.F. Mundy	GSA	O	X	X
J.K. Nelson	Western Union	M	X	X
F.W. Smith	Western Union	A		
H. Spielman	RCA	M	X	X
H. Tholstrup	Friden	M	X	
L.R. Turner	NASA	M	X	X
H.C. Umstead	USAELRDL	A		
R.E. Utman	J.C. Penney, Co.	M	X	
E.F. Vidro, Jr.	Xerox Corp.	M	X	X
A.L. Whitman	ITT Communications Systems	M		
M.A. Woodbury	N.Y. University	O	X	X

RECEIVED BEMA/DPG
JUL 27 1964

M = Member
A = Alternate
O = Observer

2. Membership Status

E.F. Manning of USAEL has resigned and B.V. Magee has been appointed his replacement. J.G. Doyle and R.R. Spalholz of Fairchild Graphic have been dropped from the membership list due to lack of attendance.

3. Correction of the Minutes for the Previous Meeting Number 12.

1. Membership Status

H. Spielman, Member from RCA, was inadvertently left off the list.

AT&T was represented by proxy.

5. Actions Taken on Document ISO/TC97/SC2 (N.Y. '64-18) 123* E

n) Mr. Kerr proposed with Mr. Lewis seconded, that position 2/0 be called Space for ISO and that for ASCII, the approved abbreviation be Sp instead of b̄.

o) Delete "with footnote (3)" following @ in the second line.

p) Delete "tilde" and insert in its place "circumflex" or "up arrow".

q) Delete first sentence and replace with "Mr. Avakian proposed, and Mr. Kerr seconded, that position 5/11 (left bracket), 5/12 (second currency), 5/13 (right bracket), 6/0 (@ symbol), 7/11, 7/12, 7/13, 7/14 respectively, be reserved for national graphic usage with footnote (3) and position 2/4 CS, with footnote (5)".

Correct the next sentence to read "The definitions that were contained in Document 123* E ....".

C. Delete the entire item and replace with "Mr. Lang proposed, and Mr. Lewis seconded, that the definition for STX be accepted for ISO and for ASCII, a character which precedes a sequence of characters referred to as text, that is to be treated as an entity, and entirely transmitted through to the ultimate destination, or which precedes a block of characters which is within such a sequence. STX terminates any heading initiated by SOH. The vote was 10 for, 1 against, 1 abstention."

4. Documents Assigned X3.2.4 Numbers

X3.2.4/

34 "A Doctrine of Code Extension for Use with the American Standard Code for Information Interchange" by D.A. Kerr, July 6, 1964.

35 "Comments on Tutorial Paper X3.3.4/44" by E. Avakian, July 10, 1964.

36 "Note on the ISO 6-Bit and 7-Bit Codes (Notes 1-5)" by E.H. Clamons, January 28, 1964.

37 "IBM's Extended BCD Card Code", by E.E. Plomondon, July 10, 1964.

38 "Revised ASCII Code Recommended by X3.2.4 to X3.2", July 14, 1964.

39 Reply to "Tutorial Paper X3.3.4/44 on Control Procedures for Data Communications, dated May, 1964" by X3.2.4 Task Group, July 15, 1964.

5. It was proposed by E.H. Clamons that the EIA Specification, RS-244, Code for Numerical Tool Control, not be included on the agenda for this meeting. This was accepted by a vote of 11 for, 0 against, and 1 abstention.

6. Program of Work Scheduled for Revision of ASCII was Established as follows:

   a. Draft code table, 7/14.

   b. Receive approval in principle for the Draft Code Table from X3.2, 7/16.

   c. 1st draft revisions 9/1,2 (Ad Hoc Committee Meetings)

      1) Draft of standard
      2) Appendices
      3) Expository remarks

   d. Review of C by X3.2.4, 9/24.

   e. Review of redraft of C by X3.2.4, 10/21.

   f. Mail documents to X3.2 for comments, 11/1.

   g. Reply on the documents from X3.2, 11/27.

   h. Final polishing of draft proposal, 12/1.

7. Ad Hoc Committees Established to Draft:

   a. Standard

      W.F. Emmons, Chairman
      J.G. Griffis
      J.L. Little

   b. Appendices

      D.A. Kerr, Chairman
      E.S. Ferguson
      L.R. Turner

   c. Expository Remarks

      J.B. Booth, Chairman
      O.R. Arne
      E.A. Avakian

8. A revised ASCII Code Table Recommendation was prepared and accepted by the Task Group, X3.2.4, by a unanimous vote.

REVISED CODE TABLE RECOMMENDATION OF X3.2.4

	0	1	2	3	4	5	6	7
0	Null	DLE	Sp	0	—	P	@	p
1	SOH	DC_1	!	1	A	Q	a	q
2	STX	DC_2	"	2	B	R	b	r
3	ETX	DC_3	#	3	C	S	c	s
4	EOT	DC_4 (Stop)	$	4	D	T	d	t
5	Enq	Nack	%	5	E	U	e	u
6	Ack	Sync	&	6	F	V	f	v
7	Bell	ETB	'	7	G	W	g	w
8	BS	Cncl	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SS	*	:	J	Z	j	z
11	VT	Esc	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	|
13	CR	GS	-	=	M	]	m	}
14	SO	RS	.	>	N	^	n	¬
15	SI	US	/	?	O	`	o	Del

An amendment was proposed to interchange the symbols I and \ locations 5/12 and 7/12. This was defeated by a vote of 2 for, 9 against, 2 abstentions.

9. A draft to the "Tutorial Paper X3.3.4/44" was prepared for X3.2 Chairman's signature and submittal for action to the X3.2 Group.

10. The second day was spent on discussing the problems and aspects of code extension. A motion was made by E.H. Clamons and seconded by D.A. Kerr "In reaffirmation of the principles of previous work on code extension, paper X3.2.4/34 is accepted to be the basis for preparation of a standard doctrine of code extension for use with ASCII". This was accepted by a vote of 11 for, 0 against, 0 abstentions.

11. A X3.2.4 Work Program on code extension was established:

Priority

1. Doctrine and Allocation of Code Extension Sequences
2. Shift-Out (SO) - Shift-In (SI)
3. Special Sequence (SS), By-Pass and Restore
4. ASCII, Control-Free and Other Sets
5. Transparency
6. Specific Extensions for ASCII Code Table
7. Review Standard Reset-Restore

12. Next meeting of X3.2.4 will be at the Hilton Inn, San Francisco, September 22 and 23, 1964.

E.F. Vidro, Jr.

E.F. Vidro, Jr.
Acting Secretary

X3.2/142
July 16, 1964

X3.2.4/38
JULY 14, 1964

REVISED ASCII CODE RECOMMENDED BY X3.2.4 TO X3.2

	0	1	2	3	4	5	6	7
0	Null	DLE	Sp	0	—	P	@	p
1	SOH	DC_1	!	1	A	Q	a	q
2	STX	DC_2	"	2	B	R	b	r
3	ETX	DC_3	#	3	C	S	c	s
4	EOT	DC_4(Stop)	$	4	D	T	d	t
5	Enq	Nack	%	5	E	U	e	u
6	Ack	Sync	&	6	F	V	f	v
7	Bell	ETB	'	7	G	W	g	w
8	BS	Cncl	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
10	LF	SS	*	:	J	Z	j	z
11	VT	Esc	+	;	K	[	k	{
12	FF	FS	,	<	L	\	l	|
13	CR	GS	—	=	M	]	m	}
14	SO	RS	.	>	N	^	n	—
15	SI	US	/	?	O	`	o	Del

CBI 67/3/2

364

Data Processing Group

bema

Business Equipment Manufacturers Association

235 East 42nd Street, New York, N.Y. 10017 – MUrray Hill 7-5969

RECEIVED BEMA/DPG

JUL 22 1964

SUBCOMMITTEE X3.2 - CHARACTER SETS AND DATA FORMATS
OF
ASA SECTIONAL COMMITTEE X3 - COMPUTERS AND INFORMATION PROCESSING

MINUTES

Meeting #39, July 16, 1964
Minneapolis, Minn.

SUMMARY

1. The Subcommittee recommended that EIA Standard RS-244 not be approved as an American Standard.
2. The proposed X3.2 scope was accepted, subject to the removal of the word "control" in the first line.
3. A revised ASCII code table was accepted as the basis for a draft revision of ASCII.
4. Task Group X3.2.3, Punched Cards, is to direct their efforts towards a "Hollerith" compatible representation of ASCII and an ASCII compatible representation.
5. Three industry comments have been received since the publication of the Proposed Recorded Magnetic Tape Standard.
6. A number of paper manufacturers attended the meeting of Task Group X3.2.2, Punched Tape, which was devoted to the gathering of data for a proposed physical paper tape standard.
7. Task Group X3.2.5, Data Formats, Related Sets and Applications, continued their study of formats, collating sequence and extended sets.
8. The letter from DOD, relating to keyboards, was judged to be an X4 matter.

NMAH
316/5

PRIVATE ASA X3 COMMITTEE DOCUMENT

NOT FOR PUBLICATION

APPENDIX A

Attendance

Name	Company	Status
J. F. Auwaerter	Teletype Corp	M
J. B. Booth	Teletype Corp	A
R. M. Brown	DOD	M
C. E. Mackenzie	IBM	M
J. R. Lakin	IBM	O
R. F. Ewald	BTL	A
D. A. Kerr	BTL	A
O. R. Arne	Mergenthaler Linotype	M
H. L. Gross	NCR	A
H. Tholstrup	FRIDEN	M
W. F. Huf	Navy	O
M Woodbury	N. Y. Univ.	O
E. S. Fergusson	Rockford Research	O
	N. Y. U. Consultant	
J. W. Sage	US Army (SCC) Comp Syst Dir	O
R. E. Mullendore	CENSUS	A
C. E. Schultheiss	Burroughs Corp	O
T. R. Bousquet	Honeywell	M
George Murphy	Ampex	M
Jim Taunt	GE	M
Emik A. Avakian	Teleregister, Div of Bunker-Ramo	M
H. Spielman	RCA	M
R. E. Short	Kleinschmidt	O
R. F. Mundy	GSA	O
E. H. Clamons	Univac	M
J. K. Nelson	W.U	M
R. M. Gryb	AT&T	M
L. L. Griffin	Navy	Chairman
J. W. Windhorst	CDC	M
J. L. Little	NBS	M
W. E. Brainard	Kearney & Trecker Corp	O
H. Ankeney	Giddings & Lewis Machine Tool Co	O
H. S. White	DOD	A
H. C. Umstead	USAEL	O
E. F. Vidro, Jr.	Xerox	M
V. L. Thompson	UNIVAC	O

IBM Technical Newsletter

File Number	File No. S360-01
Re: Form No.	A22-6821-1
This Newsletter No.	N22-0155
Date	September 18, 1964
Previous Newsletter Nos.	None

CHANGES TO IBM SYSTEM/360 PRINCIPLES OF OPERATION, FORM A22-6821-1

The attached information is to be used in place of the information currently applying to the Extended Binary Coded Decimal Interchange Code (EBCDIC). The chart, which is to replace Figure 11, shows bit positions, bit patterns, hole patterns(card), graphic characters, and control characters for the EBCDIC.

In addition, the chart for EBCDIC in Appendix F, page 141, should be deleted.

national Business Machines Corp., Product Publications Dept., Poughkeepsie, N. Y.

IN U. S. A.

Bit Positions 0, 1	00				01				
Bit Positions 2,3	00	01	10	11	00	01	10	11	Digit Punches
0000	①	②	③	④	⑤ SP	⑥ &	⑦ -	⑧	
0001							⑬ /		1
0010									2
0011									3
0100	PF	RE	BY	PN					4
0101	HT	NL	LF	RS					5
0110	LC	BS	EB	UC					6
0111	DL	IL	PR	ET					7
1000									8
Zone Punches	9 12	9 11	9 0	9	9 12 0	9 12 11	9 11 0	9 12 11 0	

Bit Positions 0,1	10				11				
Bit Positions 2,3	00	01	10	11	00	01	10	11	Digit Punches
0000					⑨	⑩	⑪	⑫ 0	8-1
0001	a	j			A	J	⑭	1	1
0010	b	k	s		B	K	S	2	2
0011	c	l	t		C	L	T	3	3
0100	d	m	u		D	M	U	4	4
0101	e	n	v		E	N	V	5	5
0110	f	o	w		F	O	W	6	6
0111	g	p	x		G	P	X	7	7
1000	h	q	y		H	Q	Y	8	8
1001	i	r	z		I	R	Z	9	9
Zone Punches	12 0	12 11	11 0	12 11 0	12	11	0		

Bit Positions 0,1	00				01				
Bit Positions 2,3	00	01	10	11	00	01	10	11	Digit Punches
1001									8-1
1010			SM		¢	!	⑮	:	8-2
1011					.	$	,	#	8-3
1100					<	*	%	@	8-4
1101					(	)	_	'	8-5
1110					+	;	>	=	8-6
1111					|	¬	?	"	8-7
Zone Punches	9 12	9 11	9 0	9	12	11	0		

Bit Positions 0,1	10				11				
Bit Positions 2,3	00	01	10	11	00	01	10	11	Digit Punches
1010									8-2
1011									8-3
1100									8-4
1101									8-5
1110									8-6
1111									8-7
Zone Punches	12 0	12 11	11 0	12 11 0	9 12 0	9 12 11	9 11 0	9 12 11 0	

① 12-0-9-8-1
② 12-11-9-8-1
③ 11-0-9-8-1
④ 12-11-0-9-8-1
⑤ No Punches
⑥ 12
⑦ 11
⑧ 12-11-0
⑨ 12-0
⑩ 11-0
⑪ 0-8-2
⑫ 0
⑬ 0-1
⑭ 11-0-9-1
⑮ 12-11

FIGURE 11. EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE

Control Characters

PF	Punch Off
HT	Horizontal Tab
LC	Lower Case
DL	Delete
RE	Restore
NL	New Line

BS	Backspace
IL	Idle
BY	Bypass
LF	Line Feed
EB	End of Block
PR	Prefix

PN	Punch On
RS	Reader Stop
UC	Upper Case
ET	End of Transmission
SM	Set Mode
SP	Space

Special Graphic Characters

¢	Cent Sign
.	Period, Decimal Point
<	Less-than Sign
(	Left Parenthesis
+	Plus Sign
|	Vertical Bar, Logical OR
&	Ampersand
!	Exclamation Point
$	Dollar Sign

*	Asterisk
)	Right Parenthesis
;	Semicolon
¬	Logical NOT
-	Minus Sign, Hyphen
/	Slash
,	Comma
%	Percent
_	Underscore

>	Greater-than Sign
?	Question Mark
:	Colon
#	Number Sign
@	At Sign
'	Prime, Apostrophe
=	Equal Sign
"	Quotation Mark

Examples	Type	Bit Pattern Bit Positions 01 23 4567	Hole Pattern	
			Zone Punches	Digit Punches
PF	Control Character	00 00 0100	12 -9	4
%	Special Graphic	01 10 1100	0	8 - 4
R	Upper Case	11 01 1001	11	9
a	Lower Case	10 00 0001	12 -0	1
	Control Character, function not yet assigned	00 11 0000	12 - 11 - 0 -9	8 - 1

Appendix G

Report of Meeting No. 14, ASA Task Group X3.2.4
Code Assignments
September 22 and 23, 1964
Hilton Inn, San Francisco, California

1. Attendance

Members and Attendees	Affiliation	Status	Sept. 22	Sept. 23
J.L. Little, Chairman	National Bureau of Stand.	M	X	X
J.F. Auwaerter	Teletype Corp.	M		
E.A. Avakian	Teleregister	M		
J.B. Booth	Teletype Corp.	A	X	X
D.A. Carruthers	Bell Telephone of Canada	O		X
E.H. Clamons	Univac	A	X	X
C.J. Davis	IBM	M		
R.J. Donahue (proposed mem.)	Mergenthaler Linotype	O	X	X
W.F. Emmons	IBM	A	X	X
R.F. Ewald	BTL	A		
E.S. Ferguson	Rockford Research Insti.	O	X	X
J.G. Griffis	DCA	M		
R.M. Gryb	AT&T	M		X
G.W. Hawkins	IBM	O	X	X
J.E. Heins	Kleinschmidt	O	X	X
W.F. Huf	Navy	M		
J.H. Hughes	Bolt Beranck & Newman	O	X	X
R.M. Ireland	SCM	O	X	
J.D. Keen	Univac Div.	M		
D.A. Kerr	Bell Telephone Labs	A	X	X
W.Y. Lang	BTL	M		
E.J. Lewis	ITT - DISD	M		
B.V. Magee	USAEL Ft. Monmouth	M	X	X
R.F. Mundy	GSA	O	X	X
J.K. Nelson	Western Union	M		
F.W. Smith	Western Union	A	X	
H. Spielman	RCA	M	X	
H. Tholstrup	Friden	M		X
L.R. Turner	NASA	M		
H.C. Umstead	USAELRDL	A		
E.F. Vidro, Jr.	Xerox Corp.	M	X	X
H.S. White, Jr.	DOD	O	X	X

M = Member   A = Alternate   O = Observer

NMAH 316/5

2. Membership

O.R. Arne (Mergenthaler Linotype) resigned and was replaced by R.J. Donahue. Mr. Donahue was considered, by the Task Group, to be an observer for this meeting only and will have full voting rights at the next meeting.

A.L. Whitman (ITT Communications Systems) has resigned because of his retirement.

3. Correction of Minutes of the Previous Meeting No. 13

Item 9, page 4: Insert the following words after draft "letter relating to".

4. Documents Distributed to X3.2.4/

40 - Capitalization of Control Abbreviations; D.A. Kerr to Members X3.2.4, August 21, 1964.

41 - ASCII Appendices; September 4, 1964 (one part of three).

42 - Draft Revision of ASCII; September 10, 1964. (Cover letter with another part of 41.)

43 - No title; E.H. Clamons to X3.2.4, September 22, 1964.

44 - ASCII Tape and Printing Symbols; August 13, 1964.

45 - Conflicts in the Usage of ASCII Controls; W.F. Emmons to Members X3.2.4, September 18, 1964.

46 - Definition of "Transparency"; W.F. Emmons to Members X3.2.4, September 21, 1964.

47 - Some Notes on Customer Requirements for Code Insensitive Transmission Channels; E. Stuart Fergusson to Members X3.2.4, undated.

48 - Suggestions to X3.2.4 on Desired Changes to X3.2.4/41, and Additional Expository Remarks to X3.2.4/41 Eight-Bit Representation; J.E. Taunt to J.L. Little, September 21, 1964.

49 - Graphic Symbols for Problem Definition and Analysis; L.L. Griffin to Chairman X3.2.4, September 2, 1964.

50 - Second Draft Revision of ASCII (Only the revised standard section was completed. Appendices and expository remarks sections will be completed at a later date.)

51 - Location of Second Currency Symbol (CS_2) in ASCII, W.F. Emmons to Members X3.2.4, September 23, 1964.

52 - IBM's Extended BCD Card Code as Revised 9/4/64; E.E. Plomondon to X3.2.4 Members, September 21, 1964.

5. Name Change for Task Group X3.2.4

It was suggested that Task Group X3.2.4 be called "Code Development". This new title will be placed before X3.2 for approval.

6. Program of Work for 1965

Priority	Title
1	ASA X3.4-1963 (ASCII) Revision.
2	Complete any necessary action on international code for information interchange.
3	Code extension doctrine and applications.

No unusual expenses are anticipated for travel or work by the Task Group for the Year - 1965.

7. The document X3.2.4/49 (X3.6/14 Working Paper on Graphic Symbols for Problem Definition and Analysis) was discussed. It was decided that the Chairman of X3.2.4 should reply to this paper with emphasis on symbols, definitions, and the working paper format which is that of a proposed standard.

8. D.A. Kerr moved that the code table typography shown in the 4 September 1964 Table (Attachment 1) be retained for the time being, with the understanding that we may wish to suggest to the ISO that they may wish to consider the elimination of 4-letter abbreviations, in which case we would do the same. This motion was passed 5 for, 0 against, 1 abstention.

9. It was suggested by L.L. Griffin that X3.2.4 might desire to change the "logical not" (¬) to down arrow (↓). A discussion followed. It was proposed by J.B. Booth that X3.2.4 retain the "logical not" in the position 7/14. This passed with a vote of 6-0-0.

10. Documents X3.2.4/48 were discussed in detail. These papers suggested that the following be included within the body of the revised code standard:

a. preferred collating sequence.
b. ASCII code within an 8-bit environment.
c. hierarchical relationship for information separators (FS, most inclusive; GS; RS; and US, least inclusive).

The first was included in the body of the standard under Qualification: "The relative sequence of any two characters, as a basis for collation, is defined by their binary values". The third item was included under the definition of File Separator, Group Separator, Record Separator, and Unit Separator in the body of the standard.

A lengthy discussion followed on the subject of ASCII code in an 8-bit environment. It was finally resolved by Mr. Kerr's motion that "the proposed material on 8-bit representation not be included in the proposed revision to ASCII at this time". This passed by a vote of 5-2-1. Mr. Clamons requested a roll call vote on this issue.

In Favor	Against	Abstention
J.B. Booth	E.H. Clamons	H. Spielman
D.A. Kerr	W.F. Emmons	
B.V. Magee		
F.W. Smith		
E.F. Vidro, Jr.		

11. Revisions to Document X3.2.4/41; Draft Revision of ASCII

a. E.H. Clamons moved that position 2/2 quotation mark (diaeresis); position 2/7 apostrophe, closing single quotation mark (acute accent); position 5/14 up arrow (circumflex or tilde); and position 5/15 opening single quotation mark (grave accent) followed by an asterisk (*) refers to the footnote saying the use of these symbols as diacritical signs is described in Appendices section A6.3. This was accepted by a vote of 7-0-0.

b. D.A. Kerr moved that, if a single "stop" control is required to interrupt or turn off ancillary devices, DC4 is the preferred assignment.

It was also moved the DC's be not completely defined but left in general terms.

Both passed by a vote of 5-0-0.

c. D.A. Kerr moved as a definition for Information Separators that: These information separators may be used within data in optional fashion except that their hierarchical relationship shall be: FS as most inclusive, then GS, then RS, and US as least inclusive. (The content and length of a File, Group, Record, or Unit are not specified.) This passed by a vote of 7-0-0.

d. F.W. Emmons moved that the "/contraction" be removed from the second sentence of 6.1. This passed by a vote of 5-0-1.

e. All the changes made to the Draft Revision of ASCII (X3.2.4/41) are contained in the attached Document X3.2.4/50, September 23, 1964.

12. Document X3.2.4/43 was discussed. Sections 1 and 2 will be considered for inclusion in the Appendices, Section B.2, when this portion of X3.2.4/41 is reviewed and rewritten.

13. Document X3.2.4/51 was discussed but no official action was taken.

14. Work was begun on the revision of the Draft Appendices Document X3.2.4/41. The review and rewriting of this document was not completed and has been tabled until the next meeting.

15. The next meeting of X3.2.4 will be in New York City at BEMA Headquarters, October 19, 20, and 21.

E. F. Vidro, Jr.

E.F. Vidro, Jr.
X3.2.4 Secretary

Attachment 1

DRAFT REVISED 4 SEPTEMBER 1964

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII)

b7					0	0	0	0	1	1	1	1
b6					0	0	1	1	0	0	1	1
b5					0	1	0	1	0	1	0	1
Bits	b4 ↓	b3 ↓	b2 ↓	b1 ↓	COLUMN → / ROW ↓	0	1	2	3	4	5	6	7
	0	0	0	0	0	NULL	DLE	SP	0	_	P	@	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
	0	1	0	1	5	ENQ	NACK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYNC	&	6	F	V	f	v
	0	1	1	1	7	BELL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CNCL	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SS	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FS	,	<	L	\	l	|
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	¬
	1	1	1	1	15	SI	US	/	?	O	`	o	DEL

X3.2

ATTENDANCE X3.2
MEETING SEPTEMBER 24, 1964
HILTON INN - SAN FRANCISCO, CALIF.

	NAME	ORGANIZATION	MEMBERSHIP STATUS
✓	L. L Griffin	Navy Department	Chairman
✓	J. E. Taunt	GE	M
✓	J. B. Booth	Teletype	A
✓	D. A. Kerr	Bell Telephone Labs	A
✓	R. J. Donahue	Mergenthaler Linotype Co.	M(proposed)
✓	R. M. Ireland	SCM	M
no	J. E. Heins	Kleinschmidt, Div SCM	O
✓	H. S. White	Department of Defense	A
✓	F. W. Smith	Western Union Tel. Co.	A
✓	E. F. Vidro, Jr.	Xerox Corp	M
no	R. R. Brookman	Burroughs Corp.	A
✓	J. L. Little	NBS	M
✓	E. H. Clamons	UNIVAC	M
no	D. A. Carruthers	Telephone Association of Canada	O
✓	R. M. Cryb	A T & T	M
no	H. C. Vinstead	USAEL	O
✓	R. M. Brown	DoD	M
✓	R. E. Mullendore	Census	A
✓	V. L. Thompson	UNIVAC	A
✓	R. F. Mundy	GSA	O
no	M. L. Ryherd	NCR	O
no	C. A. Crandall	NCR	O
✓	G. L. Murphy	Ampex	M
✓	H. Spielman	RCA	M
✓	J. Windhorst	CDC	M
✓	H. Tholstrup	Friden Inc.	M
✓	J. R. Lakin	IBM	O
✓	C. E. Mackenzie	IBM	M
already sent —	J. F. Auwaerter	Teletype	M
✓	E. W. Kettnich	Teletype	O

NMAH
310/5

X3.2/165
October 30, 1964

MINUTES OF MEETING #40 - SUBCOMMITTEE X3.2
September 24, 1964

Presiding Officer: Mr. L.L. Griffin, Chairman
Time : 8:00 a.m. to 5:40 p.m.
Place : Hilton Inn - San Francisco

RECEIVED BEMA/DPG
OCT 30 1964

1. Attendance

See Appendix A for membership list and attendance.

2. Minutes

The minutes of Meeting #39 were approved after being modified as follows:

1. In Item 9.1, "X3.2.3" was changed to "X3.2.4".
2. In Item 6, 4th Paragraph, "....in which the members of X3.2" was changed to "....in which members of X3.2...".

3. Membership

Messrs. Bob Arne, Abe Hassan, and Allen Whitman resigned. Mr. Donahue (Mergenthaler Company) was nominated for membership on X3.2 and X3.2.4. Mr. Jess Van Over (G.E.) was nominated an alternate on X3.2, and a member of X3.2.3. Eric Clamons moved that Allen Whitman be made a consultant to X3.2.

Passed 14-0-0.

4. Agenda

See Appendix B for revised agenda.

5. Documents Distributed at Meeting

The following documents were distributed at the meeting:

Document	Subject	Author
X3.2/145	Minutes of X3.2.5 Meeting	X3.2.5
X3.2/146	X3.2.5 Program of Work	X3.2.5
X3.2/147	Proposed American Standard Recorded Magnetic Tape for Information Interchange	X3.2.1
X3.2/148	Expository Remarks (supporting X3.2/147)	X3.2.1
X3.2/149	International Work on Magnetic Tape Stds.	X3.2.1
X3.2/150	Comments on Interim Federal Spec. WT 0051	X3.2.1
X3.2/151	ASCII Code Table 4, September, 1964	X3.2.4
X3.2/152	Draft Revision of ASCII	X3.2.4
X3.2/153	Program of Work X3.2.1	X3.2.1
X3.2/154	Program of Work X3.2.3	X3.2.3

ıg, Berlin, 1950.
alysis Using Special Tea-
A; l 1960, pp. 580-600.
l Engineer, October 1961.
s T nsfer (in press).

cal unrealizable. He
n, with fixed parameters,
rval, and gives examples

ı, s into methods of
ıte ocesses in pulsed
necessarily rectangular.
Each of these papers
an ainstaking presen-
lm makes a textbook

smoothing of non-linear
fo d oscillations, two
sol on of a particular
ı, a one on optimum
limitation. The diffi-
ventions in terminology
er. which is entitled
l d gn of circuits and
igh-speed servo-mech-

on the theory of
w h must constitute
at of the dynamics
d which includes an
ıse-plane methods in

nt ality. Misprints
be intolerable. The
d in the wording, but
ipli ed mathematical
ers pts of tiny size
uct

R. H. Tizard.

Computer Journal
vol 7 #3
Oct 1964

The I.S.O. character code

By H. McG. Ross*

A specific and fully detailed proposal has been prepared within the International Organization for Standardization on a character code for use in computer, data-processing and telecommunication systems. Although it is still subject to formal ratification, it is clear that this new character code and its repertoire of symbols should be considered for use in any future system. This affects designers of equipment and programming languages, systems planners, and those who have to choose the symbols and equipment used in commercial applications.

Introduction

For several years work has been in progress on developing standard character codes for use in computer, data-processing and communication systems. Ultimately, this has been under the auspices of the International Organization for Standardization, and during a recent meeting agreement was reached at the technical level on specific proposals. These have to go through a fairly long process of formal ratification, and it is impossible to forecast the outcome; but the overriding attitude of those taking part in the technical discussions was that the time has come to settle on these proposals, and thus to be able to begin to derive the benefits of using a common unified system.

The opportunity is presented, therefore, to enter a period of stability with respect to character codes; to escape from the present babel; to establish means for different users more readily to communicate with each other and interchange work; and for the development of a whole series of peripheral machines which, in this respect, are compatible and really suited for data-processing work.

The work covers three aspects: the choice of graphic symbols and control characters; the system for encoding them in binary form; and consideration of methods of exploiting the features of the overall code. Throughout the project very great care has been given to making the scheme as versatile and as widely useful as possible.

History of the project

This type of character code originated within Ferranti Limited (Mr. I. V. Idelson and Dr. C. H. Lindsey). It was first used on the punched tape for Orion and Atlas computers. This code is essentially a 6-bit code with two shifts, and when implemented on 7-track punched tape has the normal arrangement of using one track as a parity check on the data tracks. Similar character codes are used within Atlas and Orion and the various peripheral machines, and the final result is a group of eight minor variants or shifts; these are all closely related and conversion between one and another is easily affected, but it serves as an illustration of the limitations which are experienced in practice of any 6-bit code when used for anything other than the simplest equipment.

The work was developed within the British Standards Institution and in due course published as B.S. 3480: "Data processing codes for punched tape." This specification covers a series of character codes, all on punched tape, which are essentially derived from a basic 6-bit code. In the most sophisticated form, 8-track tape, the 8th track is used to determine two shifts, one containing capital letters and the other small letters. A number of British computer systems, particularly KDF 9 and Elliott 503, used punched tape and internal character codes based on B.S. 3480. The basic concepts of this type of code, as it then stood, were described in this *Journal* (Ross, 1961); most of these concepts have carried right through the project.

This type of code was used also in B.S. 3635: "Codes for punched tape for numerically controlled machine tools." As far as the character code and its implementation on punched tape is concerned, the importance of this standard lies in attempting to correlate the code for machine-tool control with that for computers. There are some small differences between this code and that of B.S. 3480 due to the developments in the international field referred to below.

This basic work was considered by workers in America (Mr. J. F. Auwaerter, and Bemer, Smith and Williams, 1961) who were developing advanced telecommunications systems—particularly those in which communication equipment and computers would be used together. Many valuable contributions were made in the American work, the more important being:

(*a*) A substantial number of characters were introduced for the control of telecommunication systems and for the associated equipment.

(*b*) The concept was introduced that the character code should be regarded in a generalized way for the interchange of data between one user or machine and another. Thus it has an abstract form, each printable symbol or control character being represented by a pattern of binary digits. Then, as a supplementary process, these are implemented on various media and in various machines, in different ways as appropriate. As a corollary, the basic code represents the form in which it is transferred from one user to another, and it is at this stage that the standard must be honoured; thereafter, each individual user is free

---

* *With Standard Telephones and Cables Ltd., Connaught House, Aldwich, London, W.C.2; formerly with I.C.T. (Ferranti Computer Department).*

to modify it in his own equipment as he wishes, thus giving substantial flexibility to designers of computers and peripheral machines.

(*c*) Major emphasis was placed on a code with 7-bit characters because, on the one hand, the limitations of any 6-bit code were clearly recognized and, on the other hand, there was a greater willingness to accept the consequences of handling the larger characters.

(*d*) A sophisticated and powerful arrangement was introduced for relating 6 and 7-bit codes.

This work, done within the various national standards organizations, passed as is normal practice to the International Organization for Standardization. The purpose of ISO is to produce internationally acceptable recommendations on which each national standards organization bases its national standard; the latter carry the ultimate authority, but this process ensures that the various national standards are identical or at least compatible. A number of other countries joined in this work, and in due course a first and then a second draft ISO proposal were produced. These proposals covered two codes in closely related form, with 6 and 7-bit characters.

The European Computer Manufacturers Association recognized the importance of having a common character code for use with the various computers in Europe, and in 1963 published a standard. Primary interest lay in a 6-bit code, and this ECMA standard follows the ISO 6-bit version. In addition ECMA published a very valuable discussion of the philosophy of this type of code, and the relation between the 6 and 7-bit forms.

The American Standards Association published in 1963 the American Standard Code for Information Interchange (ASCII). In this case primary interest lay in a 7-bit code, and therefore ASCII follows the 7-bit form of the ISO second draft proposal.

During 1963 C.C.I.T.T. – the standards organization for all international telecommunication matters – studied the ISO proposal with a view to its use for future message telecommunication systems. It was recognized that there was a need for a larger character code than the International Telegraph Alphabet No. 2—commonly called the Telex code—for future telecommunication systems and traffic although, because of the immense amount of Telex equipment currently established, the new code would complement, not displace, the Telex code. The C.C.I.T.T. introduced a particular need for adequate provision of accents and other diacritical signs which are required for any system which is to cope adequately with the many languages of the industrialized countries, even though so much of the previous computer work had been based on English.

Further study of the foregoing work, especially in telecommunications and in those fields in which IBM have exceptional experience, led to a number of suggestions for further improvements which, during May 1964, were embodied in a ISO third draft proposal. It is this which forms the basis of the present paper.

Present status of the ISO code

When a project has been worked out at the technical level, the normal procedure within ISO is for it to be ratified by the various national standards organizations. This usually takes some time, and it has to be gone through in connection with this character code, the whole matter must therefore be treated with a certain reserve until this has been completed. Then national standards may be created or modified in accordance with the ISO recommendation. B.S.I. have already announced their intention to develop B.S. 3480 to conform with the final ISO code, and it is of course to be hoped that ASCII will be similarly amended. ECMA have under discussion a 7-bit code to complement their published 6-bit code.

U.I.C. and I.C.A.O., the international organizations dealing with rail and air transport respectively, are known to be considering the use of this type of 7-bit character code for interchanging messages. An interesting development is that this character code, together with an extension to accommodate a further 64 of the special Japanese graphic symbols, has been found satisfactory for use in the Japanese computer industry.

ASCII is used in an ambitious new telecommunications service provided by the American Telephone and Telegraph Co., and, together with an extended BCD (punch-card oriented) code, it is adopted for the IBM System/360.

Explanation of ISO 7-bit code, and purpose of the characters

The ISO character code comprises two systems, one with 7-bit characters and 128 combinations, the other with 6-bit characters and 64 combinations. These are shown in Tables 1 and 2 respectively. The codes will be explained in terms of the 7-bit set, since this contains the 6-bit set.

Within each character the bits can be given the usual binary significance, the convention being to show the most-significant bit on the left. The code table is presented in a series of columns and rows, the columns being identified by the more-significant bits in each character, and the rows by the less-significant.

Considering the 128 characters as one continuous string in ascending order of significance, the characters are grouped together so that on the one hand they may be readily identified either by program or equipment, and on the other hand they collate satisfactorily for sorting. In general, for the latter purpose the sequence is controls, important separating and punctuation symbols, numerals, capital letters, and small letters; some of the less-commonly used symbols, particularly those not often involved in sorting, are interposed into gaps within this general sequence.

More often, however, advantage will be taken of the grouping into columns within the code table. For example, the control characters are identified by the two most-significant bits being 00, and for all letters

Table 1: 7-bit code table

b_7					0	0	0	0	1	1	1	1
b_6					0	0	1	1	0	0	1	1
b_5					0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7
0	0	0	0	0	(TC_0) Null	(TC_7) DLE	Space	0	—	P	(@) ③	p
0	0	0	1	1	(TC_1) SOH	DC_1	!	1	A	Q	a	q
0	0	1	0	2	(TC_2) STX	DC_2	" ⑧	2	B	R	b	r
0	0	1	1	3	(TC_3) ETX	DC_3	# ⑥	3	C	S	c	s
0	1	0	0	4	(TC_4) EOT	DC_4 (Stop)	CS_1 ⑤	4	D	T	d	t
0	1	0	1	5	(TC_5) Enq	(TC_8) Nack	%	5	E	U	e	u
0	1	1	0	6	(TC_6) Ack	(TC_9) Sync	&	6	F	V	f	v
0	1	1	1	7	Bell	(TC_{10}) ETB	' ⑧	7	G	W	g	w
1	0	0	0	8	FE_0 (BS)	Cancel	(	8	H	X	h	x
1	0	0	1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE_2 (LF) ①	SS	*	: ②	J	Z	j	z
1	0	1	1	11	FE_3 (VT)	Escape	+	; ②	K	([) ③	k	③
1	1	0	0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	(CS_2) ③	l	③
1	1	0	1	13	FE_5 (CR) ①	IS_3 (GS)	–	=	M	(]) ③	m	③
1	1	1	0	14	SO	IS_2 (RS)	.	>	N	^ ⑦⑧	n	③
1	1	1	1	15	SI	IS_1 (US)	/	?	O	`	o	Delete

Explanatory notes about the 7-bit code table

1. The controls CR and LF are intended for printer equipment which requires separate combinations to return the carriage and to feed a line.
   For equipment which uses a single combination (called New Line) for combined carriage return and line feed operation, NL will be coded at FE_2. The use of New Line requires agreement between the sender and the recipient of the data.
2. If 10 and 11 as single characters are needed (for example, for Sterling currency subdivision), they should take the place of "colon" (:) and "semi-colon" (;) respectively.
3. "Reserved for National Use." These positions are primarily intended for alphabetic extensions. If they are not required for that purpose, they may be used for symbols and the recommended choice as shown in parentheses.
5. "Reserved for National Use." A currency sign will be assigned to this position.
6. The number sign (#) in position 2/3 may have an alternate graphical representation (N°).
7. It is acceptable to represent tilde (~) by circumflex (^) for international interchange of information in Spanish and Portuguese. In Spanish and Portugese speaking countries the tilde may replace the circumflex in position 5/14.
8. The graphics in positions 2/2, 2/7, 5/14 have the significance of quotation mark, apostrophe and upwards arrow, respectively; however, these characters take on the significance of diaeresis, acute accent and circumflex diacritical signs when they follow the Backspace code.

Table 2: 6-bit code table

b6 →					0	0	1	1
b5 →					0	1	0	1
b4	b3	b2	b1		0	1	2	3
0	0	0	0	0	F_0 Space	0	Null	P
0	0	0	1	1	F_1 (HT)	1	A	Q
0	0	1	0	2	F_2 (LF) ①	2	B	R
0	0	1	1	3	F_3 (VT)	3	C	S
0	1	0	0	4	F_4 (FF)	4	D	T
0	1	0	1	5	F_5 (CR) ①	5	E	U
0	1	1	0	6	SO	6	F	V
0	1	1	1	7	SI	7	G	W
1	0	0	0	8	(	8	H	X
1	0	0	1	9	)	9	I	Y
1	0	1	0	10	*	: ②	J	Z
1	0	1	1	11	+	; ②	K	([ ) ③
1	1	0	0	12	,	< ④ CS_1	L	(CS_2) ③
1	1	0	1	13	–	= ④ %	M	( ]) ③
1	1	1	0	14	.	> ④ &	N	Escape
1	1	1	1	15	/	'	O	Delete

Explanatory notes about the 6-bit code table

1. The controls CR and LF are intended for printer equipment which requires separate combinations to return the carriage and to feed a line.
   For equipment which uses a single combination (called New Line) for a combined carriage return and line feed operation NL will be coded at F_2. This requires agreement between the sender and the recipient of the data. F_5 will then be regarded as Backspace (BS).
2. If 10 and 11 as single characters are needed (for example the Sterling currency subdivision), they should take the place of "colon" (:) and "semi-colon" (;) respectively.
3. "Reserved for National Use." These positions are primarily intended for alphabetic extensions. If they are not required for that purpose they may be used for symbols and the recommended choice is shown in parentheses.
4. Either of the two sets of symbols shown in these positions in the table may be chosen; but this requires agreement between the sender and the recipient of the data.

bit-7 is 1. Another important group is the "dense graphic sub-set" comprising the four centre columns, in which the two most-significant bits differ from each other; this contains the graphic symbols regarded as the most important for data-processing work. For example, high-speed line printers might be confined to this character set. Capital and small letters differ only with respect to bit-6; this is one of the many points taken into account to simplify the design of associated keyboards.

Certain positions in the code table are shown as available for national use; these are intended for special letters, accents or other symbols which may be needed within a particular country but not for international purposes. These characters should only be used when there is some prior understanding between the sender and recipient as to what they mean. A particular example of this concerns the currency symbols: that in position 2/4 is regarded as the primary symbol within any one country, e.g. £ in U.K., or $ in the U.S.A.; if a second currency symbol is required it should be coded at 5/12.

The symbol ≠ means the same as No., and it can be very useful. If a particular user requires a symbol not in the standard, it is a convention that the positions at the foot of column 5 (and perhaps 7) should first be used for such variants, but they require agreement between sender and recipient.

It is the control characters which are the most important and perhaps novel. There are certain groups of these, for example the Transmission Controls which are intended primarily for use in telecommunication systems, the Format Effectors which are used to control the layout of printed or other documents, the Device Controls which may be used to start and stop ancillary machines, and the Information Separators. The latter represent an important but difficult concept: they may be used to determine the structure of data, that is to say the way it is split up into large or small sections, or alternatively built up from small units into large. They comprise a hierarchy of increasing power; thus for example a *Space* might be used to separate the words within a name, the *Unit Separator* to separate items within an entry (as in the fields within a punched card), the *Record Separator* to separate each composite entry, and the *Group Separator* to identify the end of a series of entries.

In a number of controls the meaning is shown within parentheses. Then that meaning is regarded as being an example for a particular system or type of equipment, but it is permitted, for other systems, to use an alternative but related meaning. A particular example is 0/9 which means *Horizontal Tabulate* in a typewriter-like machine but *Skip* in a punched-card machine. This equivalence of the controls is specially discussed in the ECMA Philosophy document (1963).

The individual controls will now be described in general terms, in contradistinction to the formal definitions in the official documents.

Null "means nothing" (this is a very tricky thing to define precisely). It will be used as an idling condition to fill out time, or to fill out media as, for example, the leaders and trailers of a punched tape.

The next four characters are used to control and demarcate messages sent over transmission networks. *Start of Heading* indicates the beginning of an address or other preamble which may be used to establish the routing through the transmission network; *Start of Text* (STX) indicates the beginning of a message in the ordinary sense, and this is finished by *End of Text*; there may, of course, be any number of messages within one transmission, and the whole sequence is terminated by *End of Transmission*.

Enquiry is a general request for a remote station to indicate its identity or condition. The response may be an identification signal or it may be a positive *Acknowledge* or a *Negative Acknowledge* (NACK). These may be used not only to establish identity, but also to signal whether the remote station is in working condition, and they may also be used in connection with error-control systems.

Bell is to attract the operator.

Within the Format Effectors, *Backspace* means move back the *printing* position one space; if it is being punched on tape it is punched just as any other character, and it is therefore *not* to be used to move the tape backwards.

Horizontal Tabulate corresponds to the normal tab action of a typewriter, moving to the next pre-set stopping position; and *Vertical Tabulate* gives the same facility with regard to moving the paper. *Form Feed* is used with pre-printed stationery and means move the paper to the first printing position on the next form. *Line Feed* moves the paper a unit distance and *Carriage Return* moves back to the left margin; if, however, a single character or key operation is needed for this combined operation (as on electric typewriters) *New Line* is coded at 0/10.

Five shift characters are provided. *Shift In* means that the following character(s) are as in the standard. The others are used to get out of the standard code in various ways and for various purposes, and in every case the meaning of the subsequent characters would require agreement between sender and recipient. *Shift Out* would be used to go into a different complete code, perhaps a new set of graphics. *Escape* and *Data Link Escape* (the letter confined for use within transmission systems) are in general intended to allow a *single* following character to be given a special meaning. *Start of Special Sequence* (SS) is used to indicate a sequence of characters which have special significance or require special handling, especially for transmission control; this sequence would be finished by *Shift In*.

The *Device Controls* are used to start and stop ancillary machines such as readers and punches.

Sync, which has a special bit-pattern, is used to synchronise transmission systems.

End of Transmission Block may be used with certain methods of error control in transmission systems to signify the end of a block (chosen to suit the design of the transmission system) of data, and that an error control character follows.

Cancel, on the other hand, is a character which can be included in data and means that the preceding block of data contains an error and may have to be ignored. There must be agreement between sender and recipient as to how far back to go.

End of Medium may be used to identify the end of the used portion of a medium such as punched or magnetic tape.

Delete is used particularly with punched tape to overpunch errors, being the all-holes character.

The 6-bit code

The 6-bit code of Table 2 represents a selection from the 7-bit table of the characters considered most important. The general concept for the relation between the two codes may be illustrated in terms of the conversion from the 7 to the 6-bit code, as follows. The characters in the lower half of column 0 are moved to the upper half (by inverting bit-4); then the columns are collapsed by ignoring bit-6 and moving bit-7 into the 6th position. An important consequence of this technique is that the small letters "map over" the capital letters. There are, however, some exceptions to these rules of conversion, the most important being for *Backspace* and *Escape*; and for some of the alternate symbols shown at the foot of column 1 of the 6-bit code table it may be necessary also to invert bit-4. In practice, therefore, it may be best to carry out such conversion by computer program or table look-up rather than by conversion hardware.

Use of Backspace

Certain proposals for the use of the Backspace character warrant special mention because of their influence on systems design and the writing of input programs.

It is visualized that Backspace would be used to prepare accented letters, composite symbols, and for underlining as in ALGOL. In all these cases the convention would be that the letter or ordinary symbol would be keyed (and therefore punched or transmitted) first, then the Backspace character, and then the diacritical sign. By diacritical sign is meant an accent, the underline symbol (4/0) or, for example, the solidus which might be used to make not-greater-than. This triplet of characters (interspersed of course with Delete) would need to be treated as an entity. Although some criticism has been made of the complication resulting from this use of Backspace, it is small compared with the very versatile use of Backspace which is common to all Orion input routines.

A subtle point about the ISO code is that some of the symbols will be considered as accents when used in this way with Backspace, but will have another meaning when used on their own: for example, diaeresis or quotation mark, acute accent or apostrophe, and circumflex accent or upwards arrow.

Implementation of the code on various media

The basic codes being regarded in an abstract sense, work is in progress to establish standards for their implementation on the following media:

(*a*) punched paper tape
(*b*) punched cards
(*c*) edge-punched cards
(*d*) magnetic tape for data interchange
(*e*) telecommunication systems
(*f*) keyboards
(*g*) printers

On *punched paper tape*, the tracks are identified by being numbered 1, 2, 3, feed hole, 4, 5, 6, 7 and 8 (if used). Each character in the code is punched in one row across the tape, the bit numbers corresponding to the track numbers. In addition the next higher numbered track is used for a parity bit: for the 7-bit code, the parity is made even overall, and for the 6-bit code odd overall. Prime emphasis is being placed on the use of 1 inch wide tape. This is the obvious thing to use for the 7-bit code but is rather more unconventional for a 6-bit code, in which case parity goes in track 7 and the 8th track is unpunched. This is part of a move to discourage the need for future tape machines to exist for both 7/8ths and 1 inch widths.

On *punched cards* consideration is confined to cards with 80 columns and 12 rows (identified from top to bottom as *A*, *B*, 0, 1. . .9) and rectangular holes. It is emphasized that the use of punched cards with this character code in no way mitigates against their continuing use with any of the well-established card codes, such as that in B.S. 3174: "Alpha-numeric punching codes for data-processing cards," and it is to be anticipated that the two will co-exist for a substantial time.

Considerable support has been given for a very subtle arrangement whereby the ordinary single hole punching in rows 0 to 9 for the numerals is retained. With this arrangement card-rows *A*, *B* and 0 are used to identify the columns of the code table as follows:

ISO column #	0	1	2	3	4	5	6	7
Hole punched in card-row #	*A*	*A*		None	*A*		*A*	
	B	*B*				*B*		*B*
	0		0				0	0

and card-rows 1 to 9 are used to represent the row code table as follows:

ISO row #	Hole punched in card-row #
0	None
1	1
.	.
.	.
.	.
9	9
10	8 + 2
.	.
.	.
.	.
15	8 + 7

In addition, an exception to these rules is made transposing the punching for *Space* and *zero*. gives a hole in card-row 0 for zero, and also the desirable feature of an unpunched column for Sp (There may ultimately be other transpositions for spe purposes.)

For *edge-punched cards*, although no specific propo have yet been made, it is anticipated that the punch would be equivalent to a length of punched tape alon long-side of the card, with row 1 and bit 1 neare edge, and a feed-hole track between tracks 3 and 4

Implementation of the code on *magnetic tape* pre much more severe problems than the other media although substantial progress has been made the wor not yet complete. The following main aspects hav be covered:

(*a*) Dimensions and quality of the unrecorded ta
(*b*) The dimensions and other properties of the sp
(*c*) The layout and magnetic parameters of recorded tracks.
(*d*) The format of the recorded information wi respect to the way the character is placed wi the tracks and the way the record is broken into blocks, and the means for checking.
(*e*) Any labels or heading and terminating reco which may be needed to allow a compute recognize and correctly interpret the recorded d

Work on the other media is not yet sufficiently vanced to warrant a useful summary.

References

"American Standard Code for Information Interchange (1963)." *A.C.M. Communications*, Vol. 6, p. 422.

BEMER, R. W., SMITH, H. J., and WILLIAMS, F. A. (1961). "Design of an Improved Transmission/Data Processing Co *A.C.M. Communications*, Vol. 4, p. 212.

"Bit Sequencing of ASCII in Serial-by-Bit Data Transmission" (1964). *A.C.M. Communications*, Vol. 7, p. 333.

European Computer Manufacturers Association (1963). *ECMA Standard for a 6-Bit Input/Output Character Code.*

European Computer Manufacturers Association (1963). *EMCA Philosophy on Codes.*

International Organization for Standardization (1964). "Third Draft Proposal on ISO 6 Bit and 7 Bit Codes." Wor document ISO/TC 97/SC 2/125.

"Perforated Tape Code for Information Interchange" (1964). *A.C.M. Communications*, Vol. 7, p. 336.

ROSS, H. McG. (1961). "Considerations in Choosing a Character Code for Computers and Punched Tapes," *The Com Journal*, Vol. 3, p. 202.

by the link standing at the top of this stack, and the stack to be popped up. Conditional statements are written in a form reminiscent of ALGOL with, however, round brackets instead of the words **begin** and **end** to enclose compound statements. F is the free list. It is hoped that with these few words of explanation the routine will prove comprehensible.

```
10
$A$ = CAR $M$, POP UP $M$
$D$ = CAR $M$, POP UP $M$
IF CAR $A$ = ATOM THEN
(IF CAR $A$ = "$X$" THEN CAR $D$ = "1" OTHERWISE
      CAR $D$ = "0", RETURN)
OTHERWISE
($B_1$ = CAR $A$, $B_2$ = CDR $B_1$, $B_3$ = CDR $B_2$
CAR $D$ = $F$
$L_1$ = CAR $D$, $L_2$ = CDR $L_1$, $L_3$ = CDR $L_2$
$F$ = CDR $L_3$, CDR $L_3$ = "0")
IF CAR $B_1$ = "+" or "−" THEN
(CAR $L_1$ = CAR $B_1$
PUSH DOWN $M$, CAR $M$ = $L_2$
PUSH DOWN $M$, CAR $M$ = $B_2$
PUSH DOWN $M$, CAR $M$ = $L_3$
PUSH DOWN $M$, CAR $M$ = $B_3$
TO 10 AND BACK, TO 10 AND BACK
RETURN) OTHERWISE
(CAR $L_2$ = $F$
$R_1$ = CAR $L_2$, $R_2$ = CDR $R_1$, $R_3$ = CDR $R_2$
$F$ = CDR $R_3$, CDR $R_3$ = "0"
CAR $L_3$ = $F$
$S_1$ = CAR $L_3$, $S_2$ = CDR $S_1$, $S_3$ = CDR $S_2$
$F$ = CDR $S_3$, CDR $S_3$ = "0"
CAR $L_1$ = "+", CAR $R_1$ = ".", CAR $S_1$ = "."
CAR $R_2$ = CAR $B_2$, CAR $S_3$ = CAR $B_3$
PUSH DOWN $M$, CAR $M$ = $S_2$
PUSH DOWN $M$, CAR $M$ = $B_2$
PUSH DOWN $M$, CAR $M$ = $R_3$
PUSH DOWN $M$, CAR $M$ = $B_3$
TO 10 AND BACK, TO 10 AND BACK
RETURN)
```

Acknowledgement

This paper is based on an expository lecture given at the 1964 Meeting of the Association for Computing Machinery, held in Philadelphia. The paper was originally published in the Proceedings of that meeting, and I am grateful to the Association for permission to reprint.

References

McCarthy, J. *et al.* (1962). *LISP 1.5 Programmer's Manual*, M.I.T. Press.

Newell, Allen (Ed.) (1961). *Information Processing Language-V Manual*, The RAND Corp.

Wilkes, M. V. (1964). "An experiment with a self-compiling compiler for a simple list-processing language," *Annual Review of Automatic Programming*, Vol. 4, Pergamon Press, Oxford.

Wiseman, N. E. (1964). "Application of list-processing methods to the design of interconnections for a fast logic system," *The Computer Journal*, Vol. 6, p. 321.

Computer Journal Jan 1965
v7 #4

Correspondence

The Editor,
The Computer Journal,

Sir,

May I make two comments on the article "The ISO character code" by H. McG. Ross, in the October *Journal*? Firstly, I am sorry to see that Backspace will be used "to prepare composite symbols and for underlining as in ALGOL." Although this use of backspace is common, it is far inferior to the use of a non-escaping underline, as anyone who has experienced both systems will testify. The labour of punching ALGOL programs is greatly reduced by the provision of a non-escaping key with vertical bar and underline, as on the MC-ALGOL Flexowriter used at the Mathematical Centre, Amsterdam.

Secondly, I would propose the perhaps heretical view that standardization of character codes is not as important as is sometimes made out. Code translation is an easy process for a computer to carry out, and the computing system should be designed to deal with any code, translating by software or a software/hardware combination to or from a common internal code. An essential corollary to such a system is the acceptance of the printed record as authoritative: a particular representation on tape or cards may be imposed by the limitations of peripheral equipment or transmission systems, but this is not relevant to the user, and he should not have to bother about it. The user should be able to say "print a letter A" or "tell me what the next character of the input stream is," leaving it to the system to sort out the details of the physical representation. Such facilities can be provided by a sophisticated software system, and they can be provided now, with existing peripheral machines, not at some undetermined date in the future when we have all been standardized.

Yours etc.,
D. W. Barron.

The University Mathematical Laboratory,
Corn Exchange Street,
Cambridge.
9 November 1964.

Oct 6-9 1964

TELETYPE CORPORATION

5555 TOUHY AVENUE SKOKIE, ILLINOIS 60078
PHONE 676-1000 AREA CODE 312 TWX 910-223-3611

J. F. AUWAERTER

X3.2/168
November 9, 1964

TO: ALL MEMBERS AND ALTERNATES OF X3.2 AND ITS TASK GROUPS

Attached is the ISO liaison report on the October 1964 CCITT meeting I promised you. We have it promptly through the courtesy of Vince Grey of ASA.

John Auwaerter
X3.2 International Representative

JFA:LGP

Att.

TELETYPE

CBI 67/3/3

ISO/TC 97, Computers
and Information Processing

ISO/TC 97(Secretariat-33)66
November 2, 1964

Report on Meeting of CCITT Working Group on Alphabets held October 6-9, 1964 in Geneva, Switzerland

- L. Durand, representing ISO/TC 97/SC 2
- G. Willibald, representing ISO/TC 97/SC 6

SUMMARY:

The discussion of the proposed ISO 7 bit code as a basis for a new CCITT alphabet accommodating telegraphy and data transmission requirements resulted in a CCITT alphabet proposal.

CCITT tends to favor a more strictly defined code table taylored to their known system needs.

Specifically, it was recommended to:

1.) Interchange the graphic symbols "underline" and "grave accent."

2.) Reglement the use of positions for national use.

3.) Give a unique assignment to all positions for graphics.

4.) Define closer the control characters, especially the escapes.

It is felt that for the sake of a full compatibility between the ISO and CCITT code proposals, the difference mentioned in 1. should be resolved.

Introductory matters

Mr. Faugeras, Chairman of the Study Group, opened the meeting by recapitulating the work done in the past within CCITT and pointed out that with the formulated character set requirements, solutions can be found, either by use of 6-bit systems with shift or 7-bit alphabet. The latter is being given preference (already at previous meeting) mainly because present practice shows the inconvenience of frequent shift procedures. If limited character sets are to be used, they should be regarded as subsets of a 7-bit alphabet. The group then agreed to consider the 7-bit ISO proposal first.

Mr. Durand, representating ISO/TC97/SC2, then explained the status of the third draft proposal (document N°125), which is due for correspondence voting, and pointed out that the delegates present at the New York May 1964 meeting hoped that there should be no further drafts. However, comments are expected, and their nature will determine whether revisions are necessary.

The third draft, Mr. Durand said, is an improvement of the second in particular in that "alternatives" were reduced, "national positions" were increased in number and control characters were confined in columns 0 and 1. The control characters reserved for data transmission were selected and labelled.

More work has to be done in the definition field, and it is believed in ISO that implementation studies will help. For the time being, definitions are to be considered as outside of the Recommendation. CCITT has an important function to play here in defining how data transmission controls should be used in their particular field.

Mr. Durand mentioned the strong ISO and ECMA position on the advantages of using backspace with diacritical signs and concluded in hoping that a common code (or alphabet in the CCITT terminology) will be

...

agreed upon in order to permit the transmission of data over systems of mixed nature without intermediate transcoding.

The following agenda was then proposed by the Chairman, and accepted :

1. Study of ISO 7-bit Code Table.
2. Study of definitions for control characters.
3. Bit order of transmission and parity checking in serial systems.
4. Implementation in parallel systems - Track ordering in punched tape.
5. Study of possible subsets.

New documents were acknowledged and discussed at some length :

a) An IBM document introduced ideas on the "Control procedures" and the possibilities of using "sequences of characters" beginning by Escape or DLE for making extra controls. It was remarked that the new alphabet with its 128 Combinations was studied with the hope of avoiding "multi-character sequences", although the "prefix method" is easier to implement than the presently recommended sequences of 4 letters.

b) A document presented by the German Delegation pleads in favour of an alphabet limited to one set of letters. Such goal, as a separate study, cannot be looked for when the 5 bit existing alphabet provides roughly the same service and when, as said above, a limited version of a 7-bit system can be envisioned.

c) A document presented by ITT makes a plea in favour of a systematic use of multi-character sequences for the better protection of the controls against errors. The Chairman recommended strongly to consider error protection as a topic separate from the alphabet, as ISO has done.

...

1. Study of the 7-bit Code Table

1.1. α and β characters

Such characters are required when a synchronous transmission takes place in the middle of an arythmic channel. It was, however, decided not to allocate specific combination to these functions in order to preserve the "transparency" of the channel.

Solutions still have to be found, for instance by adding an extra bit onto the synchronous path. It is also admitted that in some cases existing characters such as : Null, Sync or Delete could be used.

1.2. Diacritical signs

The German Delegation supported by the Swiss delegates presented the case of the accented letters made by "non spacing diacritical signs". The Chairman showed the advantages of the ISO solution where diacritical signs are normal characters preceded by "backspace".

In support of this solution which appears to be general Mr. Besseyre reminded the Working Group that the alphabet studied is not only destined for pure message transmission on low speed channels and that arriving to a non exclusive common alphabet is worth sacrifices.

- In addition, it is recognised that backspace is a useful function in particular to "underline" text.

It was also noted at this point that underlining a complete line will be made easy if CR and LF operations are kept separate. It was then suggested to prohibit the use of NL for data transmission.

After a general agreement for having implemented Backspace, the German Delegation withdrew its proposition for non spacing characters and a unanimous decision was reached.

...

- Referring to "tilde", the Portuguese Delegation indicated that both the "circumflex accent" and the "tilde" are used in their language although it might be tolerable to use a single diacritical in international communications. It was later decided to introduce "tilde" in preference to CS2 in position 5/12 in accordance with previously stated requirements of CCITT.

1.3. Non-Roman Alphabets

- The Delegation from USSR explained their desire to have 31 letters in both upper and lower case. For international communication, they could sacrifice accents, but would regret the loss of the "underline" sign. They, therefore, strongly recommend the interchange of "underline" with "grave accent", so that "underline" would line up with "delete" and leave a neat double group of 31 combinations for letters.

  Although arguments in favour of "underline" at the top of column 4 were given, the value of the Russian proposal was recognised and the modification adopted. The ISO representative expressed regrets that such an argumentation was not presented earlier to ISO, either by a Sovietic Delegation or via CCITT.

- The Japanese Delegation distributed a national draft proposal showing "kata-kana" characters obtainable by means of a shift-out procedure. This proposition did not lead to discussions but showed the interest of having SO and SI characters.

1.4. Monetary symbols

A very long discussion took place about CS1 and CS2, which were generally considered as very dangerous in international traffic. In order to avoid confusion, it was proposed to fix the assignments of the "Dollar sign" and the "Pound sign", or to suppress them completely.

The data processing industry representatives explained that they are obliged to maintain at least one position for a currency symbol that customers require for closed systems.

...

The problem was then to materially prohibit the use of such variable symbols on public communication networks without conflicting with data processing needs.

The following solution, presented by the Italian Delegation, was then adopted : a unique new graphical symbol will represent CS1 in all non specialised machines, in particular in teleprinters of the Administrations.

It might be decided at a later date that for international communications, currencies would be expressed by a few letters preceded by this particular sign. The second currency sign was no longer required.

1.5. Duodecimal figures 10 and 11

On a question of the Chairman, the British Delegation suggested to suppress the note (2) which allowed a variation to punctuation marks (:) and (;). It is understood that for telecommunication services, positions 3/10 and 3/11 will then have no alternative meaning.

1.6. Changes on the graphic part of the Code Table

Considering all the above conclusions, the Chairman proposed to modify the Code Table as follows :

- in position 2/4 CS with graphical representation
- in position 4/0 ` (grave accent) with note (3) bis
- in position 5/12 ~ (tilde) "
- in position 5/14 ^ (circumflex accent)"
- in position 5/15 _ (underline) without note
- in position 3/10‾ : (colon) without note (2)
- in position 3/11 ; (semi-colon) "

...

In addition, parentheses were deleted from all symbols shown in positions reserved for national use, thus expressing clearly that these signs will be used in countries where these positions are not required for alphabetic extension. It was then remarked by the ECMA representative that diacritical sign cannot be accompanied by a note saying "reserved for national use", as they are intended for international interchange of texts with small letters.

Mr. Besseyre suggests to show two categories of national usage positions. Only the positions following the small letters in column 7 will be strictly reserved for national use and, therefore, prohibited in international communications. All the others are primarily for international use with symbols shown in the table, but exceptions can be made with some countries after agreement of CCITT. (see attached Temp. Doc. N° 8 E/F).

New notes were drafted and the total proposal accepted.

2. Study of ISO definitions for control characters

The ISO/TC97/SC6 working paper on "Control procedures for Data communications using the proposed ISO seven-bit code" was introduced as a temporary document (N°2-E).

2.1. Format effectors : HT, LF, VT, FF, CR, Space, BS.

All the ISO definitions were briefly discussed and accepted.

However, the following remarks were made :

. There will be a need to decide how "tabulations" are predetermined.

.. There will be a need for introducing "time filler" characters after some of the mechanical movements.

These points, Mr. Besseyre said, will be considered within Commission 8 dealing with terminal devices.

. It must be clear that BS does not backspace receiving media other than printing positions and suggested the introduction of the word "only" in the definition.

...

2.2. Escaping characters : SO, SI, Esc., DLE, SS

The apparent profusion of the escaping possibilities provided in the ISO proposal needed long explanations. It was admitted by ISO that the field of application of each possibility is not outlined yet, and the CCITT was asked to submit its possible needs.

The present experience suggests that SO-SI would be useful for shifting to special alphabets such as the Japanese Kana, and that Escape might be required for making extra controls such as "Ribbon shift to red".

The Chairman suggested that, for the moment, CCITT ignores the definitions of Esc, DLE and SS, but considers these characters as transmittable.

The need for common work between ISO and CCITT in this field was emphasised by ISO representatives.

2.3. Enq. (WRU)

Delegates stated that they want to avoid confusion between the "Who Are You" function, which is used in telegraphy (and might be extended to switched telephone networks) and the "Enquiry" function which was introduced to get responses from the data transmission and data processing terminals. It was explained at **length** that both functions are required in different systems, with slightly different meanings, and if used in a mixed system the same character can provide the two different functions at two different times. However, it was decided to call this control character WRU and to modify the definition.approximately as follows :

" A transmission control character which use is reserved for obtaining a reply from the distant station. The response includes the identification and may be complemented by the type of equipment (used). During a communication (after a connexion has been established) on the switched network, or on special links, this character may have another (expanded) meaning."

...

2.4. BELL, Ack, Nack, Cncl

ISO definitions were accepted after some discussions.

2.5. SOH, STX, ETX, EOT

The new nomenclature (heading, text) should be studied by CCITT. These characters were left for future discussions.

2.6. DC_1, DC_2, DC_3, DC_4

These characters are of interest to CCITT in order to control terminal equipments, but detailed allocation is left for future standardization. A note is added to the definition in this sense.

2.7. ETB, EM

Not discussed.

2.8. IS_1, IS_2, IS_3, IS_4

These were accepted without discussion.

2.9. Sync

In order to avoid confusion with the synchronization of synchronous channels, it was decided to add to the definition : "...between the data transmission terminal equipments".

2.10. Conclusion on Device and Transmission control characters

The list of characters introduced by ISO was generally approved together with its positioning in the table. Needs for more characters may arise, and it was admitted that such needs would be covered by the use of "Escape"(or DLE). CCITT will ask ISO to preserve one, or two, columns of graphic characters to be placed behind "Escape" for this purpose.

...

3. Bit order of transmission in serial communications

A long exposé of the Chairman introduced this difficult matter and showed that it was conceivable to adopt both orders. IBM representatives pleaded in favour of "high order bit first"; ATT presented arguments for "low order bit first". ISO and ECMA explained that there was not enough significant preference within their Working Groups to make a strong recommendation, and that those bodies prefer to leave CCITT free of deciding on behalf of their own problems.

The hearing of comments and opinions of all delegations showed a vast majority in favour of "low order bit first", and a decision was taken in this sense.

If a parity bit is used, it will be transmitted last (1).

4. Implementation of the 7-bit code on punched tape

The ISO proposal was circulated, and the assignment of the bits onto the tracks was considered to be agreable to the Group as it gives a corresponding numbering between the bits and the tracks.

If a parity bit is used, it will be recorded onto track N°8.(1).

---

(1) Error protection has not been considered relevant to the Alphabet and the requirements may vary. Therefore the use of a character parity bit or any other redundancy was left for future consideration.

C.C.I.T.T.

Groupe de travail Mixte "NOUVEL ALPHABET"
Joint Working Party on "NEW ALPHABET"

GENEVE - 6 - 9 OCTOBRE 1964

Document temporaire No.8-F/E
Temporary Document No.8-E/F

8 Octobre 1964

Note 3 - Réservé pour usage national. L'utilisation de cette combinaison est proscrite en service international.

Reserved for national use. Use of this combination in international service is not allowed.

Note 3bis ⓧ - Le signal indiqué correspond à l'usage normal en service international. Cependant une administration qui désirerait l'utiliser avec une autre signification et pour son service national seulement peut le faire; elle devra alors notifier au Secrétariat Général de l'Union sa décision d'affecter ce signal à un usage national. L'utilisation de ce signal pour les relations internationales avec ce pays ne sera alors pas permise.

The signal shown in this case corresponds to the normal international use. However an administration may decide to use it with an other meaning and for its national use only; in this case, that Administration will have to notify to the General Secretariat of the Union its decision to take this combination for its national use. The use of this combination will then not be allowed in the international relations with that country.

Caractères codés pour l'échange d'information entre matériels de traitement de l'information.

Coded character sets for information processing interchange.

TABLEAU DE CODE A 7 ÉLÉMENTS D'INFORMATION

7 BIT SET TABLE

Bits												
				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1	Row / Column	0	1	2	3	4	5	6	7
0	0	0	0	0	(TC_0) Null	(TC_7) Ⓡ	Space	0	` ⓧ	P	@ ⓧ	p
0	0	0	1	1	(TC_1) SOH	DC_1	!	1	A	Q	a	q
0	0	1	0	2	(TC_2) STX	DC_2	" ⑧	2	B	R	b	r
0	0	1	1	3	(TC_3) ETX	DC_3	# ⑥	3	C	S	c	s
0	1	0	0	4	(TC_4) EOT	DC_4 (Stop)	CS ¤	4	D	T	d	t
0	1	0	1	5	(TC_5) WRU	(TC_8) Nack	%	5	E	U	e	u
0	1	1	0	6	(TC_6) Ack	(TC_9) Sync	&	6	F	V	f	v
0	1	1	1	7	Bell	(TC_{10}) ETB	' ⑧	7	G	W	g	w
1	0	0	0	8	FE_0 (BS)	Cncl	(	8	H	X	h	x
1	0	0	1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE_2 (LF)	[SS] Ⓡ	*	:	J	Z	j	z
1	0	1	1	11	FE_3 (VT)	[ESC] Ⓡ	+	;	K	[ⓧ	k	③
1	1	0	0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	~ ⓧ	l	③
1	1	0	1	13	FE_5 (CR)	IS_3 (GS)	—	=	M	] ⓧ	m	③
1	1	1	0	14	SO	IS_2 (RS)	.	>	N	^ ⓧ	n	③
1	1	1	1	15	SI	IS_1 (US)	/	?	O	___	o	Delete

Ⓡ Function reserved for revised definition

ⓧ See attached temp. doc. N° 8 E/F

Oct 6-9 1964 X3.2/192

TELETYPE CORPORATION

5555 TOUHY AVENUE SKOKIE, ILLINOIS 60078
PHONE 676-1000 AREA CODE 312 TWX 910-223-3611

J. F. AUWAERTER

January 11, 1965

TO: ALL X3.2 AND X3.2.4 MEMBERS AND ALTERNATES

Attached is report of CCITT meeting in Geneva on October 6-9, 1964.

John Auwaerter
X3.2 International Representative

JFA:LGP

Att.

TELETYPE

CBl 67/3/4

It was however borne in mind that the permanent space polarity could cause the 0/0 combination to be received if no special precautions were taken in the start-stop apparatus receiving that polarity (case of a clearing signal). The "all space" (0/0) NUL combination must therefore be left unallocated for the present.

II.2.2 Diacritic signs

The allocation of combinations for the five diacritic signs previously requested by the C.C.I.T.T. was considerably changed in the third I.S.O. draft 7-bit code. The I.S.O. representative explained the reasons for these changes to the members of the Working Party. These are based mainly on the fact that three non-diacritic signs (i.e. with normal printing progression) can also be used to accent capital or small letters; when such signs are used as diacritic signs and are printed above, to ensure that they are suitably placed by reference to the letter, the signal "BS" (Backspace) will be used, as provided for in the alphabet.

These signs are the apostrophe, upwards arrow (mathematical sign) and quotation marks, which become respectively the acute accent, the circumflex accent and the diaeresis when used as diacritic signs.

Certain delegates did not particularly favour the method of printing above with backspace for diacritic signs, and thought that the "Backspace" mechanism was more complicated than that of the forward stop with diacritic signs. The Working Party however admits that apparatus using the 7-bit code, with all its possibilities, will have to be provided with a backspace.

Should the use of purely diacritic signs be retained, discussion shows that two methods are used by different countries to accent letters :

i) the letter is first printed, then the diacritic sign. This generally means that diacritic signs are inserted in the printing mechanism, one step back, since there has been a step forward after the letter to be accented has been printed.

ii) First the accent is printed, without any forward movement, then the letter is placed underneath.

These two methods involve two types of apparatus which produce printing which does not match up, and even if a single apparatus could be designed, there might be many faults in accentuation due to acquired habits. This is therefore another reason for giving favourable consideration to the I.S.O. proposal. In the case of accentuation by using signs which are not pure diacritics, i.e. in the normal printing position, this incompatibility is avoided by using top printing. The signal "BS" (Backspace) need only be transmitted between the letter and the accent or between the accent and the letter, at the choice of the operator, to have in any case a correct printing of the accented letter.

Finally the control procedures conforming to the I.S.O. proposal were adopted, with the following observations :

In view of their position in the code table, the signal 2/2 (quotation marks) and the signal 2/7 (apostrophe), if they are used with the signal 0/8 (backspace), can provide diaeresis and acute accents for all international communications, without taking any account of special agreements concerning signals reserved for national uses. On the other hand, as the other signals can produce the grave accent, the circumflex accent and the tilde are indicated only as being characters preferred for international use, but which may be possibly replaced by characters for national use (by notification - see under II.2.6).

This arrangement cannot of course constitute an obstacle to the use of combinations reserved for national use in printing with one signal for an accented capital or small letter such as the letters ä, ö and ü for instance, in the Germanic languages. In the internal traffic of the countries concerned, this allocation would facilitate operations by avoiding the transmission of a sequence of 3 signals, including the signal BS (Backspace).

II.2.3 Special case of the sign "Tilde" (~).

The I.S.O. suggested that the circumflex accent should also represent the tilde. Following discussion, during which it was confirmed that some countries needed both signs, the Working Group preferred the two signs to be kept separate. Box 5/12 of the code table, made available by the abandonment of the second currency sign CS-2 (see II.2.5) would be allotted to the tilde.

II.2.4 Carriage return and line feed

The possibility envisaged by the I.S.O. of choosing between use of the two signals CR (Carriage Return) and LF (Line Feed) or of one signal NL (New Line) has once more been rejected.

The two signal method is more elastic, since it makes possible corrections by top printing if required, and also underlining from the beginning of the line. More important still, the suggested variant would create an incompatible situation by making communication impossible between apparatus of different designs. Communication with stations of the operating administrations in particular would be impossible for apparatus working with the signal NL.

It follows that note (I) of the I.S.O. Table should be deleted, use of the signal NL being inadmissible.

II.2.5 Currency signs

The I.S.O. draft provided for two currency signs: CS-I (square 2/4) and CS-2 (Square 5/12). It was thought to use these signs for the written Dollar ($) and Pound (£) signs. However, all that is really needed by those who use data processing systems is a sign showing that the following (or preceding) figures refer to an agreed currency unit, specified elsewhere if necessary. For communications between computers, or between a computer and a teleprinter, a general currency sign CS is therefore sufficient.

Box 2/4 is reserved for the CS signal, the graphical sign for which is ¤ . The sign CS-2 is abandoned.

II.2.6 Signals reserved for national use

The C.C.I.T.T. had requested five possibilities in capital letters and the corresponding five possibilities in small letters. The I.S.O. draft reduced these possibilities.

The Working Group proposes the following solution :

i) Transfer the grave accent from box 5/15 to box 4/0 and the underlining sign (_) from box 4/0 to box 5/15;

ii) allocate boxes 7/11, 7/12, 7/13 and 7/14 for national use only, as provided in the I.S.O. draft.

iii) the following boxes are, for the normal international service, given the following "preferential" allocations :

- box 4/0 - grave accent (`)
- box 5/11 - left square bracket ([ )
- box 5/12 - tilde (~) (Ref.II.2.3)
- box 5/13 - right square bracket ( ])
- box 5/14 - circumflex accent (^)
- box 6/0 - commercial at (@)

However, if an Administration wishes to use one of these last six combinations with another meaning, and for its national service only, it may do so, but must notify the General Secretariat of the Union of its decision to allot the signal for national use. The use of this signal for international relations with that country will not then be allowed, except by special arrangement.

II.2.7 Duodecimal numbering

The possibility of an alternative use of the combinations in boxes 3/10 for : or 10, and 3/11 for; or 11 is not being retained. These boxes are used solely for the colon (:) and semi-colon (;).

II.2.8 Tabulating signs

It is made clear that the manner in which the signals for :

- vertical tabulation (VT) in box 0/11,
- horizontal tabulation (HT) in box 0/9,
- form feed (FF) in box 0/12,

have to act on the receiving apparatus will have to be a matter for prior agreement between the users.

II.2.9 "Backspace" signal

It is specified that the use of this signal is limited to printing. It cannot control a backspace on intermediate recording devices (perforated tape, magnetic tape etc.)

II.2.10 "Shift out, shift in, Escape, Start of Special Sequence, and Data Link Escape" signals

These are the signals in boxes 0/14(SO), 0/15(SI), 1/11(ESC), 1/10(SS) and 1/0(DLE).

The description of these signals' functions in the I.S.O. draft lacks precision and is sometimes contradictory, the reason being that the standardization of them by I.S.O. is not finished. For data transmission these five functions will be needed but the role allotted to each has yet to be settled.

The Working Party notes that, for the transmission of messages, similar needs to those envisaged by I.S.O. may arise in the future. For instance, the combinations "Shift out" and "Shift in" might be used, as proposed by the Nippon Telegraph and Telephone Public Corporation, to provide a shift-lock giving the possibility of transmitting an alphabet with non-Latin characters (by agreement between the Administrations concerned or between correspondents). Such a possibility might be accepted on an international communication by prior agreement between the interested parties.

Thus the Working Party agrees to the use of boxes 0/14 and 0/15 for the signals "Shift out" and "Shift in" respectively. The description of these functions is the one given by I.S.O., but without the explanation given for their use combined with special sequences.

The uses of boxes 1/11, 1/10 and 1/0 have been reserved; their assignment will be decided later by I.S.O. It will have to be possible with the assignment to obtain an inversion which, on one or two columns of the code, will introduce supplementary signs to be standardized with a view to their use by the telecommunication services.

II.2.11 "Enquiry" signal

The details supplied by I.S.O. on the definition of the "Enquiry" function (box 0/5) show that this signal is sent as a request for a response from a remote station. The response may include the name of that station.

For operation with switching on the general network, the Working Party considers that the sending of the answer-back code (abbreviated name which may be followed by information on the type of equipment) must be capable of remote control, at least at the beginning and end of a connection. A "Who are you?" signal is therefore indispensable.

X3.2/163
October 22, 1964

BRIEF REPORT OF CCITT MEETING

WORKING PARTY ON A NEW TELEGRAPH ALPHABET

Time and Place:

Geneva, Switzerland
October 6 through 9, 1964

1. The meeting was attended by Postal Administrations and/or Equipment Manufacturers from the following countries:

   Austria
   Belgium
   Canada
   France
   Germany
   Italy
   Japan
   Netherlands
   Norway
   Poland
   Portugal
   Sweden
   Switzerland
   UK
   USA
   USSR

   as well as International Organizations listed below:

   ECMA
   ISO
   WMO
   FIEJ

2. During the meeting, the ISO Third Draft Proposal, 7 Bit Code, was reviewed and accepted with recommendations for the following minor changes:

   a. The following rearrangements were made to satisfy a Russian need for 5 national usage characters to permit two-case encoding of the 31 characters of the Cyrillic alphabet:

      i) Underline moved from position 4/0 to position 5/15

      ii) Positions 0 in columns 4 and 6 and positions 11, 12, 13 and 14 in columns 5 and 7 were assigned to national usage

      iii) Preferred use of "national usage" characters was indicated as follows:

Position	Preferred
4/0	Accent Grave (`)
5/11	[
5/12	Tilde (~)
5/13	]
5/14	Circumflex (^)
6/0	@

CBI 67/3/4

NOTE: These changes are minor but very significant since they greatly facilitate use of the code by Russian, Greek and Portugese speaking countries.

b. In the spirit of compromise, CCITT agreed to the ISO recommendation that diacritical marks be made by a 3 character sequence (e.g., e Back Space ´).

c. The second currency signal was eliminated and the first currency sign designated as an "anonymous" graphic (requiring further characters to identify the monetary unit) of the shape ¤.

d. "Alpha" and "Beta" (control characters to facilitate handling asynchronous transmission over synchronous links) will be handled outside the code set.

e. Other changes of a minor nature.

3. The ISO representative plans to report these recommendations to the ISO (perhaps noting them on the ballot on the third draft proposal).

4. On bit order of transmission, the group expressed an overwhelming view in favor of "low-order-first" transmission. They expressed appreciation of ISO/TC 97's statement that ISO recognizes the final authority of CCITT in this field. The ECMA representative reported a large majority within the ECMA technical committees also favoring "low-order-first."

5. Adopted the proposed ISO encoding of the code in paper tape, saving only the question as to whether the parity bit will be standard. In addition, it was agreed work on the sense of parity and the character structure would require more time than available at this meeting.

6. Both ISO and CCITT representatives agreed that the meeting greatly furthered the project for a common communications and data processing code alphabet.

J Auwaerter

J. F. AUWAERTER

X3.2.4/55

TO: X3.2.4

FROM: Eric H. Clamons

DATE: October 6, 1964

Excerpt relating to Diacritical Marks from Webster's New International Dictionary, Second Edition, unabridged.

accent - A mark over, under, or through a letter to indicate stress, quality of sound represented, or the like; a diacritical mark; also, the letter or a group of letters so marked to indicate a sound; also, any printing type having such a mark. Specif.: (a) A mark to indicate the nature and place of the spoken accent. (b) A mark to indicate the length or quality of sound of the letter marked, usually called graphic, or grammatical, accent; as, the French accents. In ancient Greek the acute accent (´) meant a raised tone or pitch; the grave (`) indicated that the tone was lowered from a high pitch, but probably not so far as to the level or flat tone of final unaccented syllables; the circumflex (^ , ⌒ .~;ater) designated a tone which was first raised and then depressed. The same varieties occurred in Latin, in which, however, such pitch accent was probably subordinate to stress. In modern dictionaries, spelling books, and the like, the acute accent is used to indicate stressed or accented syllables; as, ac'cent. It is used in French over e (é) to indicate the close quality (theoretically; actual practice varies) in cases where the e without accent would be mute; as Vénus, venus. It may indicate stress, as in Spanish, or a long vowel, as in Hungarian and Czechoslovak. In works on elocution and phonetics it often indicates a rising inflection of the voice, the grave indicating a falling inflection, and the circumflex (^) a compound (rising-falling) inflection. The acute is further sometimes used to show that a final e is pronounced, as sometimes in Bronté. The grave in French indicates (theoretically) the open quality, as in élève, or distinguishes meaning only, as in la, là. It is sometimes used in English print over the e of a final -ed to show that it forms a separate syllable, as in armèd. The circumflex has been sometimes used in modern times in writing Latin to indicate contraction. In modern French it regularly indicates a contraction, as in mûr, from OF. meür, from L. maturum; pâte, from OF. paste. The circumflex is variously used in modern English over vowels to indicate "length" a "broad" quality, etc.

acute accent - A mark (´) used to indicate stress, and for various other purposes.

circumflex - A mark (^ , ⌒ , later ~) orig. used in Greek over long vowels to indicate a compound (rising-falling) tone, and thence in other languages to mark length, contraction, etc., and in phonetic notation to indicate special sounds; called also circumflex accent.

diacritical - Serving to separate or distinguish; specif., distinguishing;--said of a mark, point, or sign attached to a letter or a symbol to distinguish it from another of similar form, as in Semitic aphabets, or to show that it stands for a particular sound as distinguished from its other sounds; thus ā, ä, ä, ō, ŏ, etc.

grave - (a) Characterized by the tone or quality indicated by the grave accent; marked with this accent; as, grave e (e). (b) Entirely unaccented.

tilde - A diacritical mark written over n in Spanish to form the letter ñ, which represents a simple palatal nasal, as in cañon. The tilde corresponds to a dash or straight line written instead of m or n, used often in medieval script and occasionally still in German script. The same mark, called til, represents in Portuguese the nasalization of a vowel or diphthong, as in lã, não, põe. This use of the til has been widely adopted in the transcriptions of Oriental tongues. The letter ñ is often used to transliterate a letter representing a palatalized n in Sanskrit and other Oriental languages. In transcriptions of African and Oceanic tongues the letter ñ is sometimes written for the guttural nasal sound (ng).

NMAIL 310/5

ESCAPE SYMBOLS NEEDED FOR EUROPEAN LANGUAGES USING THE LATIN ALPHABET

æ	ligature ae (as used for example in Danish)	is to be typed	/ae/	
œ	ligature oe (as used in French)	"	/oe/	
ß	ligature ss or sz (German)	"	/sz/	(not /ss/)
þ	letter "thorn" (Anglo-Saxon and Icelandic)	"	/tn/	(not /th/)
ð	letter "edh" "	"	/dh/	

The five escape symbols listed above stand for the insertion in their place of the characters which they represent. The escape symbols tabulated below, however, do not represent characters themselves; they simply direct that some diacritical mark be applied to the letter next following.

´	acute accent:	/ac/	étude	is	typed	/ac/etude
`	grave accent:	/gr/	mère	"	"	m/gr/ere
^	circumflex:	/cx/	rôle	"	"	r/cx/ole
¸	cedilla	/ce/	façade	"	"	fa/ce/cade
~	tilde	/ti/	cañon	"	"	ca/ti/non
¨	umlaut or dieresis	/um/	für	"	"	f/um/ur
¯	macron	/ma/	nō	"	"	n/ma/o
˘	breve	/br/	nŏt	"	"	n/br/ot
ˇ	haček (Czech, etc.)	/ha/	Čapek	"	"	/ha/Capek
-	bar (Polish)	/ba/	Łuck	"	"	/ba/Luck
˛	hook (Polish, etc.)	/ho/	miesiąc		"	miesi/ho/ac
˚	circlet (Swedish, etc.)	/ci/	år	"	"	/ci/ar
/	slash through letter	/sl/	første	"	"	f/sl/orste
˝	double acute (Magyar)	/2ac/	pengő	"	"	peng/2ac/o
·	dot (Lithuanian, etc.)	/do/	żiema	"	"	/do/ziema
	no dot (Turkish)	/nd/	Mayıs	"	"	May/nd/is
,	comma beneath (Latvian)	/cb/	nedeļa	"	"	nede/cb/la

Common

ê ñ

Czech

French

For up/low machines it is more advisable to place the lower case characters in the zones 4 and 5 where, because of the monocase machines, 5 places are reserved for national characters, and to place the capital letters in zones 6 and 7 where only three places will then be required for the national characters. Two places will become available for international use. This exchange does not affect the monocase versions.

(5) The signs ⓐ and # are used only in English speaking countries which, on the other hand, have no domestic letters. It is suggested that these two signs be assigned the places of the national characters.

(6) In monocase machines underligning may be omitted. This sign may then be relocated from 4/0 to place 6/0 which has become vacant.

(7) The places 4/0, 5/11-14 and 7/11-13 are made available for domestic use. Instead of place 4/0, place 5/15 may be reserved for national characters.

Incorporating the modifications suggested in points 4 to 7, a code according to Fig. 1 will result. By way of example, the assignment for the USA has been allocated to the places of the national characters. The Scandinavian countries will require å æ ø and Å Æ Ø, Germany ä ö ü ß and Ä Ö Ü, France ç é `^ ' and Ç.

(8) Backspacing involves an equipment complication which is out of proportion with the benefit derived and monocase machines should be spared this burden. BS should therefore be provided only for up/low machines.

(9) DC1 to DC4 are earmarked for on/off switching of accessories. In most cases these functions are not required. In other cases 4 characters are not sufficient. If the DC functions are accomplished with the aid of ESCAPE, their number need not be limited to 4.

X3.2.4/63

Tentative

7-BIT CODE
C.C.I.T.T. VERSION
9 October 1964

	0	1	2	3	4	5	6	7
0	(TC₀) Null	④ (TC₇) DLE	Space	0	` ①	P	@ ①	p
1	④ (TC₁) SOH	④ DC₁	!	1	A	Q	a	q
2	④ (TC₂) STX	④ DC₂	" ③	2	B	R	b	r
3	④ (TC₃) ETX	④ DC₃	# ②	3	C	S	c	s
4	④ (TC₄) EOT	DC₄ (Stop)	¤ (CS)	4	D	T	d	t
5	(TC₅) WRU	④ (TC₈) Nack	%	5	E	U	e	u
6	④ (TC₆) ACK	(TC₉) Sync	&	6	F	V	f	v
7	Bell	④ (TC₁₀) ETB	' ③	7	G	W	g	w
8	FE₀ (BS)	④ CNCL	(	8	H	X	h	x
9	FE₁ (HT)	EM	)	9	I	Y	i	y
10	FE₂ (LF)	④ SS	*	:	J	Z	j	z
11	FE₃ (VT)	④ ESC	+	;	K	[①	k	{
12	FE₄ (FF)	IS₄ (FS)	,	<	L	~ ①	l	|
13	FE₅ (CR)	IS₃ (GS)	–	=	M	] ①	m	}
14	④ SO	IS₂ (RS)	.	>	N	^ ①③	n	¬
15	④ SI	IS₁ (US)	/	?	O	_	o	Del

① "Reserved for National Use" as alphabetic extensions. If not needed may be used for symbols.

② The # may have an alternate graphical representation (N°)

③ The graphics in positions 2/2, 2/7, 5/14 have the significance of quotation mark, apostrophe and upward arrow, respectively; however, these characters take on the significance of diaresis, acute accent and circumflex diacritical signs when they follow the Backspace code

④ Reserved for further standardization, definition, or consideration.

Is this the latest USA.

NMAH 310/5